INTEGER SEQUENCES WITH GOOD AUTOCORRELATION PROPERTIES

A Thesis Submitted
in Partial Fulfilment of the Requirements
for the Degree of

MASTER OF TECHNOLOGY

By

Sqn. Ldr. B. B. JAIN

to the

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY, KANPUR

AUGUST, 1981

EE-1981-M-JAI-INT

CERTIFICATE

This is to certify that the work embodied in this thesis entitled : "INTEGER SEQUENCES WITH GOOD AUTOCORRELATION PROPERTIES" carried out by Sqn.Ldr.B.B.Jain, under my supervision has not been submitted elsewhere for a degree.

Aug.81

M.U.Siddiqi
Assistant Professor
Department of Electrical Engg.
I.I.T.Kanpur

ACKNOWLEDGEMENT

I wish to place on record my gratitude to my guide Dr. M.U.Siddiqi for his constant guidance and encouragement. Many of the ideas on programming were the results of fruitful discussions I had with Sqn.Ldr.S.Aggarwal. I thank him. Finally I wish to thank my wife Sneh who helped me in correcting and rewriting the draft.

Mr. O.N.Dikshita deserves credit for neat typing of the thesis.

CONTENTS

LIST OF FIGURES

ABSTRACT

Sequences with good autocorrelation properties have a number of practical applications in Radar, Sonar, Navigation etc. Sequences like pseudorandom, Barker and Huffman are well-known. Whereas pseudorandom and Barker sequences are integer sequences, Huffman sequences in general are complex and thus difficult to implement. Integer Huffman sequences will not only be easy to generate but also would be ideal for many applications where impulse equivalent autocorrelation may be required. Some results on synthesis of integer Huffman sequences have been recently reported by Ackroyd. In this thesis we extend these results to obtain integer Huffman sequences of various lengths using digital computer. In addition a scheme for hard-ware implementation of such sequences is suggested.

In some applications integer sequences satisfying arbitrarily specified autocorrelation may be required. Unfortunately no systematic techniques exist for generating such sequences. Using Psuedo-Boolean techniques for solving a linear equation we develop a computer algorithm by which integer sequences of specified autocorrelation may be obtained. Integer sequences upto length 32 with 5 elements $(0, \pm 1, \pm 2)$ have been generated using this technique and exhaustive lists of the following sequences are obtained.

a) Ternary Barker sequences upto length 15

b) Quinquinary Barker sequences upto length 13

c) Quinquinary broad Barker sequences upto length 13.

Finally as an application, integer Huffman sequences are used to estimate impulse response of a linear system. It is found that integer Huffman sequences give faster and more accurate results than those given by PN & Barker Sequences.

CHAPTER 1

INTRODUCTION

Sequences with good autocorrelation properties are of practical significance to Radar, Sonar, Digital Communications, Navigation and Telemetry. Some of these sequences like Barker, psuedorandom (PN) and Huffman are well known. Barker and PN sequences are integer sequences. Huffman sequences, however, may consist of complex elements. Because of ease of implementation integer sequences have an advantage over complex sequences.

The aim of this Thesis is to develop techniques for synthesizing various types of integer sequences and study the feasibility of using integer Huffman sequences for System Identification.

1.1 USES OF SEQUENCES WITH GOOD AUTOCORRELATION PROPERTIES

Application of sequences depends mainly on their autocorrelation properties. Although Barker, PN and Huffman sequences differ in various aspects, autocorrelation functions of all of them have a relatively narrow high peak at the centre with low amplitude sidelobes. To understand the significance of this property we give below several interesting examples.

a) Reliability of digital systems in Data Communications is determined by accurate synchronisation. In almost all

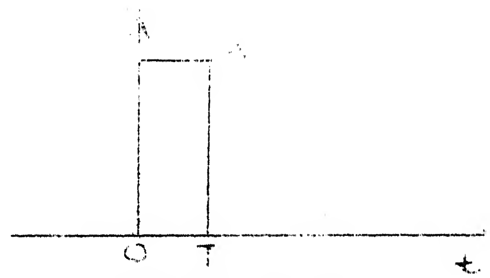
instances of practical interest, the data bit stream contains data in blocks, words or blocks of words, called frames. The start of an n bit data frame is indicated by one or more bits periodically inserted at the beginning of each frame. Sometimes only a single bit is needed for this purpose. More often, however, it is necessary to obtain frame synchronisation more rapidly or at lower SNR's than permitted using a single bit for frame synchronisation. An entire L bit code word is used for the frame sync word. Detection of frame sync is accomplished by operating on the detected binary bit stream. These synchronisation words are usually detected by a matched filter. Barker devised a method of frame sync in which the sync word is located by correlating successive L bit segments of the received bit sequence with the stored sync word. Barker used binary Barker codes for this purpose.

b) These sequences are also used as address codes on channels, where information from several data sources is to be sent simultaneously. Several receivers may be involved. Each message has an address specified by a pulse sequence, which distinguishes the source from which the message is derived and the receiver for which the message is intended.

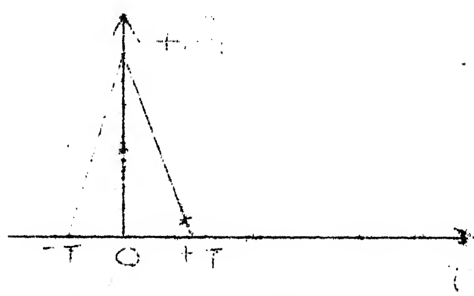
c) Application of these sequences in Pulse Compression

for use in Radar and Sonar is of great interest. This is because of the fact that target detection in the presence of white noise by correlation receiver depends only upon the energy of the signal and good range resolution requires a signal having wide bandwidth.

Consider a single narrow pulse, which has an autocorrelation function of the shape as shown in Fig 1.1. It permits very accurate determination of time of arrival of an incoming signal and thereby gives an accurate measure of range to target.



SINGLE PULSE



AUTOCORRELATION

A SINGLE PULSE & ITS AUTOCORRELATION FUNCTION.

FIG 1.1

With peak power limitations, the energy can be increased (and hence detection capability) by widening the pulse. The reduction in bandwidth is compensated by appropriate modulation of the carrier by the pulse e.g by means of coded pulse sequences. Such signals should have autocorrelation which approximates that of a single narrow pulse. This technique called Pulse Compression[3] allows tradeoff between peak power and signal duration without sacrificing time resolution. The utility of this property has also been demonstrated in the precision ranging of planetary and lunar spacecraft. Other applications such as determining altitudes for navigational purposes are possible.

d) System Identification is another area where sequences with good autocorrelation property find interesting application. Determination of impulse response of linear systems can be done with more speed and better accuracy using some of these sequences.

1.2 PN, BARKER & HUFFMAN SEQUENCES

a) PN SEQUENCES

They are binary phase shift sequences (0-180°) of length L. The usefulness of these sequences stems from their good periodic autocorrelation property. Periodic autocorrelation is defined as

$$C(K) = \sum_{i=1}^{L} x_i \; x_{i+K} \text{ (MOD L)}$$

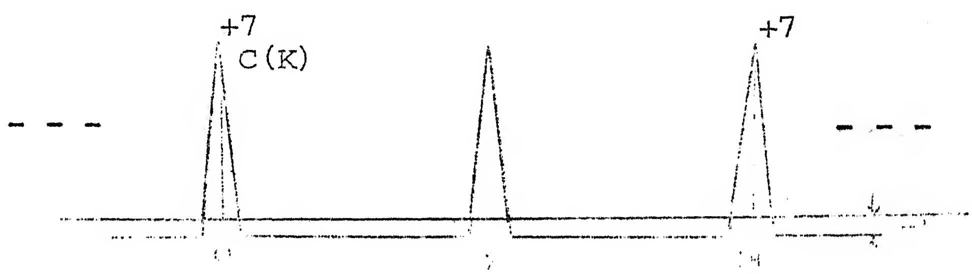
$$= L \quad \text{for} \quad K = 0, \pm L, \pm 2L\ldots$$

$$= -1 \quad \text{elsewhere}$$

where x_i's are the elements of the sequence.

C(K) is the autocorrelation at shift K and L is the length of the sequence. The number of ones per period is always one more than the number of minus ones.

A large value of L leads to two possible advantages; large peaks in the autocorrelation and a strong resembelence to a random sequence. These sequences exist only for certain values of L. For n, any integer, there is a PN Sequence with period $L = 2^n-1$. These sequences are wellknown, with easily predictable properties. They lend themselves to linear shift register generation, requiring n stages in the register. Autocorrelation of a 7 length PN sequence is shown in fig. 1.2.



AUTOCORRELATION OF A PN SEQUENCE
(L = 7,-1,-1,1,-1,1,1,1)

FIG 1.2

b) BARKER SEQUENCES

Barker sequences are again a sequence of 1's & - 1's. They possess the property of a good aperiodic auto-correlation, which is defined as

$$C(K) = \sum_{i=1}^{N-K} x_i \, x_{i+K} ,$$

where

$C(K)$ = aperiodic autocorrelation at shift K

$x_i = \left\{ \begin{matrix} +1 \\ - \end{matrix} \right\}$

N is the length of the sequence

and

$K = 0 \ldots (N-1)$

Barker sequences have

$$\begin{aligned} C(K) &= N \quad \text{for} \quad K = 0 \\ &= \{0, \pm 1\} \quad \text{for } K = 1, 2, 3, \ldots N-1 \\ &= 0 \text{ for } K > N \end{aligned}$$

Thus the sidelobes never exceed unity in magnitude, with the zero shift value only dependent on the length of the sequence. For the applications discussed earlier one would probably desire sequences of great length so as to minimize the effect of the sidelobes. The known binary

Barker sequences, however, are as shown in Table 1.1

N	SEQUENCE
2	+ +
3	+ + −
4	+ + − + , + + + −
5	+ + + − +
7	+ + + − − + −
11	+ + + − − − + − − + −
13	+ + + + + − − + + − + − +

TABLE 1.1

c) IMPULSE-EQUIVALENT PULSE TRAINS (HUFFMAN SEQUENCES)[1]

Instead of limiting the elements of the sequence to be ± 1, like in the case of Barker Sequences, if we consider real and complex numbers as the elements, we can obtain theoretically an autocorrelation function which approximates that of a single pulse as closely as possible.

This results in an aperiodic autocorrelation

$$
\begin{aligned}
C(K) &= E \qquad \text{For } K = 0 \\
&= 0 \qquad \text{For } K = 1, 2 .. N-2 \\
&= J \qquad \text{For } K = N-1 \\
&= 0 \qquad \text{For } K \quad N.
\end{aligned}
$$

Where, value of E & J depend upon the elements.

The resulting correlation is thus exactly zero every-

where except for zeroshift and for a shift which is one less than the length of the finite sequence.

The general process of generation of these sequences can be summerised as follows[1]

The sequence of amplitudes is represented as the sequence of coefficiants of a pollynomial P where

$$P = C_o D^N + C_1 D^{N-1} + .. C_N$$

If the pollynomial Q is given by

$$Q = C_N D^N + C_{N-1} D^{N-1} + .. C_o$$

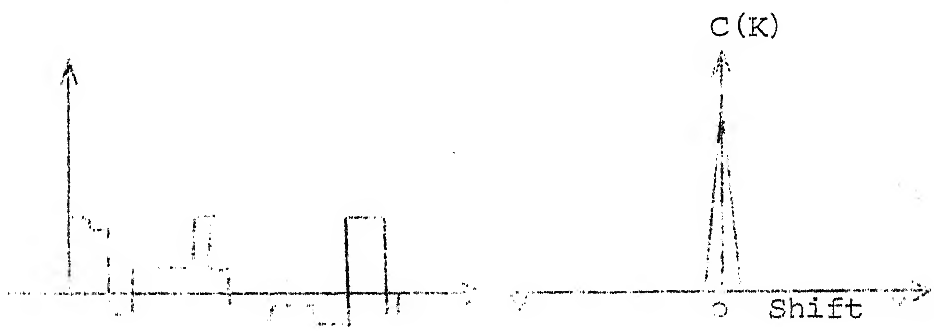
The autocorrelation of the sequence is given by the product PQ^* which is

$$PQ^* = C_o C_N^* D^{2N} + (C_o C_{N-1}^* + C_1 C_N^*) D^{2N-1} + ..$$
$$(C_o C_o^* + C_1 C_1^* + .. C_N C_N^*) D^N + .. C_N C_o^*$$

where the coefficients of PQ^* equal the autocorrelation for the corresponding shifts.

We have to choose the coefficients of P such that all coefficients of PQ^* are zero except for the coefficients of D^{2N}, D^N, & D^o. The coefficients are specified by the roots of P. For each root r_j of P there is a root $1/r_j^*$ of Q^*. It is shown that the roots of PQ^* lie on two origin centred circles in the complex plane. The specification of N of these roots of P is to be made, remembering that if a particular root of P is on the

inner circle, the other root at that angle on the
inner circle, the other root at that angle on the
outer circle is a root of Q^*. There are thus 2^N ways of selecting the roots of P from those of PQ^*. All lead to the same autocorrelation function. However since energy is not uniformly distributed here, it is reasonable to try to select that set which comes closest to having a uniform energy distribution. Fig. 1.4 shows a sketch of what could be an impulse equivalent sequence with a typical autocorrelation.



AN IMPULSE EQUIVALENT PULSE TRAIN

Fig. 1.4

1.3 ORGANISATION OF THE THESIS

Some of the integer Huffman sequences have been studied by Ackroyd[2]. In Chapter 2 we extend his technique to synthesize other integer Huffman sequences. A possible scheme for generating integer sequences using digital hardware is also described.

In some applications integer sequences with specified

autocorrelation may be required. No systematic techniques exist for the same. Moharir[4] has suggested a method using Terminal Admissibility Techniques. In Chapter 3 we rêview the method suggested by Moharir and using psuedoboolean techniques for solving a linear equation, develop a computer algorithm by which an integer sequence meeting the specified autocorrelation can be generated. Sequences with elements $\{0, \pm 1, \pm 2\}$ and lengths upto 32 have been obtained using this method and exhaustive listing of the following types of sequences with highest possible central to sidelobe ratios for each length are given.

a) Ternary Barker sequences upto length 15.

b) Quinquinary Barker sequences upto length 13.

c) Quinquinary broad Barker sequences upto length 13.

d) Integer sequences with good autocorrelation upto length 8, with elements from $\{0, \pm 1, \pm 2, .. \pm 7\}$

Finally in Chapter 4 , as an application of integer Huffman sequences, feasibility of using these sequences for system identification is studied and the results compared with those obtained using Barker and PN sequences.

Chapter 5 gives conclusions and suggestions for further work in this area.

CHAPTER 2

SYNTHESIS OF INTEGER HUFFMAN SEQUENCES

A Huffman or impulse equivalent sequence[1] is a finite sequence of complex numbers $\{C_0\ C_1 \ldots\ C_N\}$ whose autocorrelation is zero except for shifts of zero and $\pm$ N that is

$$\sum_{i=0}^{N-r} C_i\ \overset{*}{C}_{i+r}=0 \quad \text{for } r=0, \ \ldots\ N-1. \tag{2.1}$$

to be useful in various applications mentioned earlier, a Huffman sequence of length N+1 should have following two properties

a) The ratio of the amplitude of the autocorrelation central lobe to that of the sidelobe $E/|C_0\ \overset{*}{C}_N|$, where $E= \sum_{i=0}^{N} C_i^2$ should be large.

b) The energy ratio, ie. the ratio of the total sequence energy to the energy of the largest individual element, $E/\max_i /|C_i^2|$, should be large to ensure good performance in noise despite a transmitter of limited peak power. An ideal Huffman sequence, therefore, would be one which has same magnitude for all elements. Such a sequence is called Uniform Huffman sequence.

2.1 UNIFORM HUFFMAN SEQUENCES[2]

Uniform Huffman sequences are defined as those sequences for which

$$|C_0| = |C_1| = \ \ldots\ |C_N| \tag{2.2}$$

The advantage of uniform Huffman sequence would be maximum energy ratio and no necessity of a modulator. However, it is known that Uniform Huffman sequences of length greater than 3 do not exist[2] (The only uniform Huffman sequence are 1,-1 & 1,1,-1) For larger lengths, therefore, we can approximate equation (2.2) by choosing integer elements which are not widely varying in amplitudes. We can thus form integer Huffman sequences, which will have several advantages as given below.

2.2 ADVANTAGES OF INTEGER HUFFMAN SEQUENCES

a) Modulation can be very easily implemented using digital switching.

b) A digital matched filter at the receiver could be accurately implemented using integer arithmatic.

c) Integer Huffman sequences could be useful in synchronisation and Identification of Systems.

d) They could be easily generated using digital hardware. A possible scheme is suggested later in this . Chapter.

2.3 SYNTHESIS OF INTEGER HUFFMAN SEQUENCES OF ODD LENGTH.

For the purpose of our study, we can divide this into different cases depending upon the value of autocorrelation at (N-1) shifts, which is nothing but $|C_0 \ C_N|$

a) CASE (a)

$C_0 = 1 \quad C_N = -1$ Length N+1 odd

b) CASE (b)

$C_0 = 1 \quad C_N = 1$ Length N+1 odd

c) CASE (c)

$C_0, C_N \neq 1$ Length N+1 odd

Case (a) has been studied by Ackroyd. For the sake of continuity, however, it is discussed again. Case (b) and (c) have been studied in this paper. In addition computer programmes for generating integer Huffman sequences in all the three cases have also been executed and are given in appendices G to I.

2.31 CASE (a)

Two complementary ways of obtaining the sequences are possible, namely the direct solution of eqn. (2.1) and synthesis using Z transform.

2.311 DIRECT SOLUTION

The direct solution of (2.1) in integers, subject to restrictions mentioned above leads to following conclusions.

a) No solutions exist when $L = 4a+1$, $a \geq 2$ ($L=3$ being a special case)

b) For L=7 there is a class of integer Huffman sequences given by $1, 2m, 2m^2, m(m^2-1), -2m^2, 2m, -1$ where m is any integer.

c) For L=11 there is a class of integer Huffman sequences given by $\{1, 2m, 2m^2, 2m(m^2+1), 2m^2(m^2+2), m(m^4+m^2-3), -2m^2(m^2+2), 2m(m^2+1), -2m^2, 2m, -1\}$ where again m is any integer.

The derivation of further formulae for L= 15,19.. though possible becomes progressively more cumbersome. However examination of the zero pattern of the Z transforms of the foregoing sequences suggests an alternative approach to their synthesis.

2.312 SYNTHESIS USING Z TRANSFORMS

The Z transform[1] of a Huffman sequence $\{C_0, C_1, \ldots C_N\}$ is given by $C(Z) = \sum_{i=0}^{N} C_i Z^{-i}$ (2.3)

It is known that the zeros of C(Z) satisfy two conditions.

(i) The arguments of the zeros must be $(2\pi n/N) + \theta$, $n = 0, 1, \ldots N-1$ where θ is an arbitrary constant.

(ii) Each zero must lie on a circle of radius X or X^{-1} centred at the origin.

We have accordingly considered for N=6, 10, 14..., a configuration of N Zero's $Z_0 Z_1 \cdot Z_n - 1$, having the following properties.

(i) Arg. $Z_n = 2\pi n/N \quad n= 0, 1 \ldots \quad N-1$

(ii) $|Z_0| = X^{-1}$ and the remaining zero's lie at a radius of x or x^{-1} according to whether their argument is respectively an even or an odd multiple of $2\pi/N$

Fig. 2.1 shows such a zero pattern for N=14. Such a pattern is clearly the zero pattern of the Z transform of a Huffman sequence and can be regarded as consisting of 4 superimposed pole zero patterns.

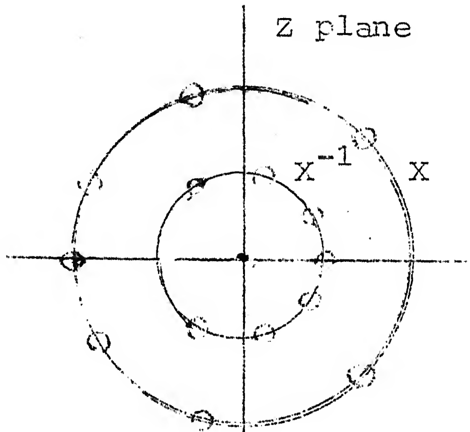


Fig. 2.2

(i) N/2 zeros situated at $Z= Xe^{j4\pi K/N}, K=0, 1 \ldots N/2-1$

(ii) N/2 zeros situated at $Z= X^{-1}e^{j2\pi(2K+1)/N}, K=0, 1, \ldots N/2-1$

(iii) A simple pole at $Z=X$ and a zero at $Z=X^{-1}$.

(iv) A simple pole at $Z=-X^{-1}$ and a zero at $Z=-X$

The poles in pattern (iii) and (iv) cancel corresponding zeros in pattern (i) and (ii). $C(Z)$ is the product of 4 factors, one factor corresponding to each of the patterns (i)-(iv). Consequently

$$C(Z)=(1-X^{N/2}Z^{-N/2})\ (1+X^{-N/2}Z^{-N/2})\ \frac{(1-X^{-1}Z^{-1})}{(1-XZ^{-1})}\ \frac{(1+XZ^{-1})}{(1+X^{-1}Z^{-1})}$$

$$=\left[1-(X^{N/2}-X^{-N/2})Z^{-N/2}-Z^{-N}\right]\frac{\left[1+(X-X^{-1})Z^{-1}-Z^{-2}\right]}{\left[1-(X-X^{-1})Z^{-1}-Z^{-2}\right]}\ . \qquad (2.4)$$

For the Huffman sequence to consist of integers, that is for integer coefficients in (2.3), we see from (2.4) that $(X-X^{-1})$ should be an integer, and $N/2$ should be an odd integer, for then $X^{N/2}-X^{-N/2}$ is expressible as a sum of powers of $X-X^{-1}$. We, therefore, choose $(X-X^{-1})=m$ where m is any integer. Equation (2.4) now can be written as

$$C(Z)=\left[1-(X^{N/2}-X^{-N/2})\ Z^{-N/2}-Z^{-N}\right]\ \frac{(1+mZ^{-1}-Z^{-2})}{(1-mZ^{-1}-Z^{-2})}\ .$$

The first $N/2$ elements of the sequence can be found as the solution of the following difference equation.

$$C_K=mC_{K-1}+C_{K-2},\quad K=3,4\ldots N/2-1,$$

where $C_0=1$, $C_1=2m$, $C_2=2m^2$.

The centre element $C_{N/2}$ is given by

$$C_{N/2} = mC_{N/2-1} + C_{N/2-2} - C_0(X^{N/2} - X^{-N/2})$$

The remaining elements can be obtained from

$$C_{N-K} = -C_K(-1)^K, \quad K = 0, 1, \ldots N/2-1.$$

Table 2.1 shows these sequences upto length 35, together with central/sidelobe ratio $E/|C_0C_N|$, the energy ratio $E/\max_i |C_i|^2$ and the efficiency $E/(N+1)\max_i |C_i|^2$. The computer programme which was used to generate these sequences is given in Appendix G

2.32 CASE (b) $(C_0=1,\ C_N=1)$

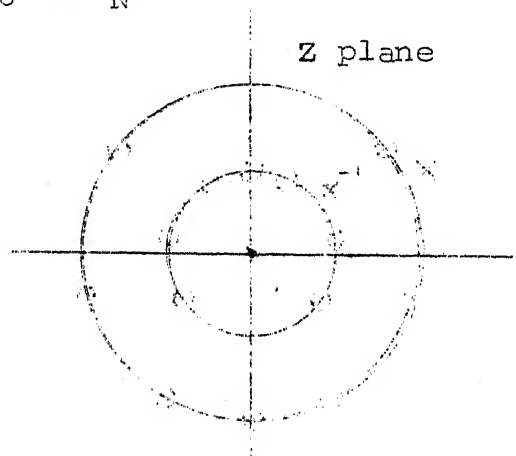


Fig. 2.1

Consider a zero pattern as shown in Fig.2.2 for N=14. It is clearly the zero pattern of the Z transform of a Huffman sequence and again can be regarded to be consisting of four superimposed pole zero patterns.

TABLE 2.1

L=N+1	m	SEQUENCE	LOBE RATIO	ENR	EFF
7	1	1, 2, 2, 0, -2, 2, -1	18	4.5	0.64
	2	1, 4, 8, 6, -8, 4, -1	198	3.1	0.41
11	1	1, -2, 2, -4, 6, 1, -6, -4, -2, -2, -1	123	3.41	0.31
	2	1, -4, 8, -20, 48, -34, 48, -20, -8, -4, -1	6726	2.9	0.26
15	1	1, -2, 2, -4, 6, -10, 16, 3, -16, -10, -6, -4, -2, -2, -1	843	3.29	0.2
19	1	1, -2, 2, -4, 6, -10, 16, -26, 42, 8, -42, -26, -16, -10, -6, -4, -2, -2, -1	5778	3.27	0.17
23	1	1, -2, 2, -4, 6, -10, 16, -26, 42, -68, 110, 21, -110, 68, -42, -26, -16, 10, -6, -4, -2, -2, -1	39603	3.27	0.14
27	1	1, -2, 2, -4, 6, -10, 16, -26, 42, -68, 110, -178, 288, 55, -288, -178, -110, -68, -42, -26, -16, -10, -6, -4, -2, -2, -1	271443	3.27	0.12
31	1	1, -2, 2, -4, 6, -10, 16, -26, 42, -68, 110, -178, 288, -466, 754, 144, -754, -466, -288, -178, - -110, -68, -42, -26, -16, -10 -6, -4, -2, -2, -1	1860498	3.27	0.1
35	1	1, -2, 2, -4, 6, -10, 16, 26, 42, -68, 110, -178, 288, -466 [illegible] 754, -1220, 1974, 377, -1974 -1220, -754, -466, -288, -178, -110, -68, -42, -26, -16, -10, -6, -4, -2, -2, -1	12752043	3.27	0.1

(i) N/2 Zeros situated at $Z= jXe^{j4\pi K/N}$, $K=0,1 \ldots N/2-1$.

(ii) N/2 Zeros situated at $Z= jX^{-1}e^{j2\pi(2K+1)/N}$,

$K=0,1 \ldots N/2-1$.

(iii) A simple pole at $Z= jX$ and a zero at $Z= jX^{-1}$.

(iv) A simple pole at $Z= -jX^{-1}$ and a Zero at $Z=-jX$.

The poles in pattern (iii) and (iv) cancel corresponding zeros in pattern (i) and (ii). $C(Z)$ is the product of four factors one factor corresponding to each pattern (i) to (iv). consequently

$$C(Z)= (1-jX^{N/2}Z^{-N/2})(1+jX^{-N/2}Z^{-N/2}) \frac{(1-jX^{-1}Z^{-1})(1+jXZ^{-1})}{(1-jXZ^{-1})(1+jX^{-1}Z^{-1})}$$

$$C(Z)= \left[1-j(X^{N/2}-X^{-N/2})Z^{-N/2}-Z^{-N}\right] \frac{\left[1+j(X-X^{-1})+Z^{-2}\right]}{\left[1-j(X-X^{-1})+Z^{-2}\right]} \quad (2.5)$$

Working on the same lines as in case (a) the solution of (2.5) can be given in an iterative form wherein

$C_1=1$, $C_2= -j2m$, $C_3= -2m^2$,

$C_K= -C_{K-2}-jmC_{K-1}$ $K= 3, N_2$,

$C_{N/2+1} = -C_{(N/2-2)}-jC_{N/2-1}+ C_0(X^{N/2}-X^{-N/2})$.

The remaining elements can be obtained from

$C_{N-K}= -C_K(-1)^K$, $K= 0,1 \ldots N/2-1$.

Table 2.2 shows these sequences upto length 31. The Computer programme for generating these sequences is given in Appendix H.

Table 2.2

N+1	m	SEQUENCE	LOBE RATIO	ENR	EFF
7	1	1, j2, 2, 0, -2, j2, 1	18	4.5	0.64
	2	1, j4, 8, j6, -8, j4, -1	198	3.1	0.41
15	1	1, j2, -2, -j4, 6, -j10, -16, j3, -16, j10, 6, -j4, -2, -j2, 1	843	3.29	0.22
	2	1, j4, -8, -j20, 48, j116, -280, -j198, -280, j116, 48, -j20, -8, +j4, 1	228486	2.91	0.19
23	1	1, j2, -2, j4, 6, j10, -16, -j26, 42, j68-110, j68, 42, -j26, -16, j10, 6, -j4, -2, j2, 1	39603	3.27	0.14
31	1	1, j2, -2, -j4, 6, j10, -16, -j26, 42, j68, -110, -j178, 288, +j466, -754, j144, -754, j466, 288, -j178, -110, j68, 42, -j26, -16, j10, 6, -j4, -2, j2, 1	1860498	3.27	0.1

2.33 CASE C $(C_o, C_N > 1)$

If the value of m is taken as p/q where p & q are integers, we can rewrite the difference equations as (From case a)

$$C_0=1\ ,\quad C_1= 2p/q\ ,\quad C_2= 2p^2/q^2\ .$$

$$C_K=(p(C_{K-1})/q)+C_{K-2}\ .$$

The Central element C(N/2) is given by

$$C_{N/2} = p(C_{N/2-1})/q + C_{N/2-2} - C_0(X^{N/2} - X^{-N/2})$$

The remaining elements can be obtained from

$C_{N-K} = -C_K(-1)^K$, $K=0,1\ldots(N/2)-1$.

Table 2.3 shows these sequences upto length 27.

TABLE 2.3

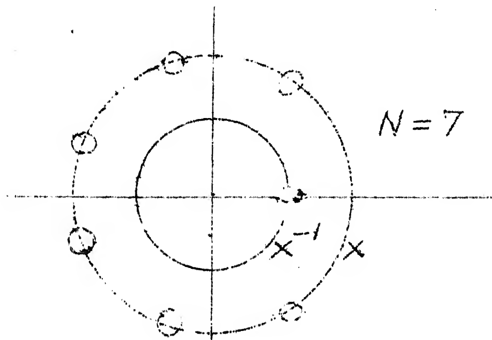
N+1	m	SEQUENCE	LOBE RATIO	ENR	EFF
7	1/2	2, 2, 1, -3, -1, 2, -2	6	3.0	0.43
11	1/2	32, 32, 16, 40, 36, -43, -36, 40, -16, 32, -32	12	6.62	0.6
15	1/2	128, 128, 64, 160, 144, 232, 260 -339, -260, 232, -144, 160, -640, 128, -128			
19	1/2	512, 512, 256, 640, 576, 928, 1040, 1448, 1764, -2363, -1764 1448, -1040, 928, -5760, 640, 256, 512, -512	86	4.03	.21
23	1/2	2048, 2048, 1024, 2560, 2360, 3712, 4160, 5792, 7056, 9320, -11716, -15843, -11716, 9320, -7056, 5792, -4160, 3712, -2304, 2560, -1024, 2048	231	3.86	0.16
27	1/2	8192, 8192, 4096, 1024, 9216, 14848, 16640, 23168, 28224, 37280, 46864, 60712, 77220, -104779, -77220, 60712, -46864, 37280, -28024, 23168, -16640, 14848, -92160, 1024, -4096, 8192, -8192	622	3.80	.141

The listing of computer programme is given in Appendix 'I'

2.4 INTEGER HUFFMAN SEQUENCES OF LENGTH 2^n

With zero patterns as envisaged in Section 2.3, integer Huffman sequences satisfying $L \neq 4a+1$, $a \geq 2$ and L = odd only, can be generated. Equation 2.4 gives an integer solution, only if N/2 is odd. That is because if N/2 is odd, L = (N+1) is also odd.

In order to generate integer Huffman sequences of even lengths, consider an alternate zero pattern as shown in Fig.2.3



Zero Pattern for N = 7, Length 2^n
Fig. 2.3

Using this pattern sequences of length 2^n, n= 1, 2....
can be generated. The pattern consists of

(i) N Zeros situated at $Z = Xe^{j2\pi K/N}$ $K=0,1... N-1$

(ii) A simple pole at Z=x and a zero at $Z = X^{-1}$

The pole in pattern (ii) cancels corresponding zero in pattern (i) Therefore

$$C(Z) = \frac{(1-Z^{-N}X^{N})\cdot(1-Z^{-1}X^{-1})}{(1-Z^{-1}X)}$$

$$= \left[1+ XZ^{-1}+X^{2}Z^{-2}+\ldots X^{(N-1)}Z^{-(N-1)}\right]\left[1-Z^{-1}X^{-1}\right]$$

$$= 1+Z^{-1}\ (X-X^{-1})+Z^{-2}(X^{2}-X^{0})+\ldots Z^{-(N-1)}(X^{N-1}-X^{N-3})$$

$$+ Z^{-N}(-X^{N-2})$$

The coefficients of the sequence are, therefore,
$1, (X-X^{-1}), (X^{2}- X^{0}) \ldots (X^{N-1}- X^{N-3}), -X^{N-2}$.

Hence

$$C_0 = 1,\ C_1 = X-X^{-1},\ C_K= X^{K}- X^{K-2} \quad K=2,\ldots N-1 \text{ and}$$
$$C_N= - (X)^{N-2}.$$

Table 2.4 shows these sequences obtained up to length 32.

TABLE 2.4

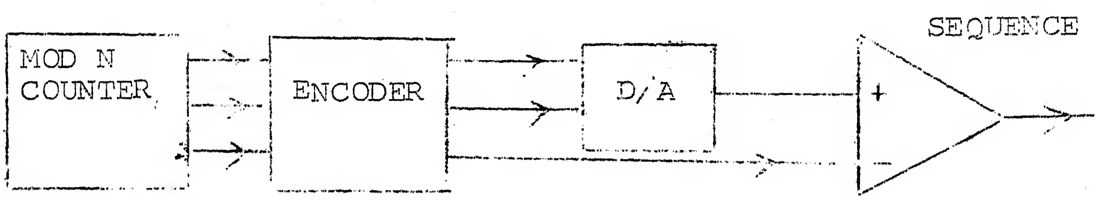
N+1	X	SEQUENCE	LOBE RATIO	ENR	EFF
4	2	4, -6, -3, -2	8	1.77	0.44
8	2	2, 3, 6, 12, 24, 48, 96, -64	128	1.77	0.22
16	2	2, 3, 6, 12, 24, 48, 96, 192, 384, 768, 1536, 3072, 6144, 12288, 24576, -16384	32768	1.77	0.11
32	2	2, 3, 6, 12, 24, 48, 96, 192, 384, 768, 1536, 3072, 6144, 12288, 24576, 49152, 98304, 196608, 393216, 786432, 1572864 3145728, 6291456, 12582912, 25165824, 50331648, 100663306, 201326792, 402653584, 805307168, 1610614336, -536870910	$.214748 \times 10^{10}$	1.77	0.06

A listing of the Programme used to generate these sequences is given in Appendix "J".

From a perusal of tables 2.1 to 2.4, it is obvious that integer Huffman sequences may not exist for all lengths. Further as the length of sequence is increased, the spread of its elements increases very rapidly. Even though central to side-lobe ratio becomes progressively higher, the efficiency falls to very low level. Hence very large integer Huffman sequences may have only limited usefulness.

2.5 GENERATION OF INTEGER HUFFMAN SEQUENCES

A possible scheme to generate an integer Huffman sequence of length 7 is described here. This scheme can be extended to generate integer sequences of larger lengths.



Block Diagram of Proposed Scheme of Generation of Integer Huffman Sequence.

FIG. 2.4

Referring to Fig.4,

mod N Counter simply counts from 0 to N-1. Its binary output x_0, x_1, x_2 is applied to an encoder. The output of the encoder is designed to be as specified in table 2.5

TABLE 2.5

Sl. no.	INPUT			OUTPUT		
	x_2	x_1	x_0	A_2	A_1	A_0
1	0	0	0	0	0	1
2	0	0	1	0	1	0
3	0	1	0	0	1	0
4	0	1	1	0	0	0
5	1	0	0	1	1	0
6	1	0	1	0	1	0
7	1	1	0	1	0	1

The output of encoder is in Binary signed magnitude form. Its magnitude is converted into an analogue voltage and the sign is attached to the magnitude through an OP AMP. The scheme can be extended to larger lengths.

CHAPTER 3

GENERATION OF INTEGER SEQUENCES OF SPECIFIED AUTOCORRELATION

In Chapter 2 certain methods were developed to generate integer sequences, that satisfied the autocorrelation of an impulse equivalent sequence. It was observed that these sequences exist only for certain lengths and that as the length is increased the uniformity in element size goes on decreasing rapidly, resulting in reduction of efficiency and energy ratio. Further it is known that Barker sequences having uniform elements $\{\pm 1\}$, exist only for certain lengths and the largest length is 13, thus limiting maximum central to sidelobe ratio to 13.

For many applications like Pulse Compression[3], Infrared Spectrometry[4], sequences with large central to sidelobe ratios with high efficiency are required. In other applications sequences with a specified autocorrelation may be required. Integer sequences, because of ease of implementation, would offer an attractive solution in many of these applications. No systematic design techniques are available to synthesize such sequences. The problem is solved either by simple enumeration or by trial and error. Simple enumeration requires a very large number of sequences to be tested, which becomes formidable even for lengths of order 16. One would, therefore, like to cut down the number of sequences to be tested.

3.1 USE OF TERMINAL ADMISSIBILITY TECHNIQUE

One approach[4] has been to use Terminal Admissibility Techniques. This technique is best explained by the help of an example.

EXAMPLE[4]

obtain all the sequences of length 16 with the **autocorrelation**

$$R(K) = \left[16, 1, 0, -1, -2, 1, -2, -1, -2, 1, -2, -1, 2, 1, 0, -1\right]$$

with $\{\pm\}$ as elements.

SOLUTION.

The required sequence may begin with 1 or -1 (written as + or - henceforth), but in view of the fact that the autocorrelation of the sequence remains unaltered by multiplying every element in it by-1, it could be assumed for the purpose of enumeration that the sequence begins with $x_0 = +$. We require that

$$R(15) = x_0 \; x_{15} = -1 \qquad (3.1)$$

Therefore

$$x_0 = + \; ; \; x_{15} = - \; .$$

The beginning of the sequence can be extended either as ++ or +- and the ending of the sequence can be extended as +- or -- . Thus there are 4 combinations of two bit beginnings and two bit endings, but it is required that

$$R(14) = x_0 x_{14} + x_1 x_{15} = 0$$

The only permissible pairs of beginnings and endings are ++, +- and +-, --. Assuming that the beginning +- is chosen, it can be extended as +- + or + - - and the permissible ending - - can be extended as + - - or - - - . Once again, there are four pairs of beginnings and endings. But as it is required that

$$R(13) = x_0 x_{13} + x_1 x_{14} + x_2 x_{15} = 1$$

only + - + and + - - or + - - and - - - are permissible pairs of beginnings and endings. This procedure can be continued recursively. The set of permissible pairs of beginnings and endings at every step, constitute the Terminal Admissibility List of that order. For example + - + , + - - , + - - , - - - is the terminal admissibility list of order 3. Terminal admissibility list of order 8 are

Beginnings	Endings
+ + + - + + + +	+ - - + - + + -
+ + + - + + + -	- - - + - + + -
+ - - + - + + +	+ - - - + - - -
+ - - + - + + +	- - - - + - - -

Further if the Terminal Admissibility Pairs are concactinated, we obtain 4 sequences of length 16, which meet the specification on R(K), K>8. The search for sequences, which meet the full specification on R(K), need be restricted to

only these 4 sequences. In this particular example only two sequences

S_1 = + + + − + + + − − − − + − + + −

and

S_2 = + − − + − + + + + − − − + − − −

out of these 4 are the required sequences. The efficiency of Terminal Admissibility Technique lies in the elimination of inadmissible pairs at successive stages.

The use of Terminal admissibility technique is limited by the fact that if the number of permissible elements is more, it becomes tedius to keep a track of various endings and beginnings. We describe below a procedure, in which we generalise this technique for larger number of elements in the set and make it suitable for computer programming.

3.2 SYNTHESIS WITH LARGER SET OF ELEMENTS

The basic problem of designing a sequence for a specified autocorrelation, lies in satisfying the set of equations

$$R(K) = \sum_{i=1}^{N-K} x_i x_{i+K} \qquad (3.3)$$

from a specified set of integers. For every value of R(K), we have to choose x_i's in a manner that eq. (3.3) is satisfied. The method is again explained by an example.

EXAMPLE

Obtain the sequences of length 7 with the autocorrelation $\left[E, 0, 0, 0, 0, 0, -1\right]$, with $\{0, \pm1, \pm2\}$ as elements. E is the value of central sidelobe.

SOLUTION

Let the sequence be termed as $(x_1, x_2 \; \ldots \; x_7)$. Autocorrelation for above sequence using eqn. (3.3) will consist of the following equations.

$$x_1^2 + x_2^2 + x_3^2 + \ldots . \; x_7^2 = E \tag{3.4}$$

$$x_1x_2 + x_2x_3 + x_3x_4 + x_4x_5 + x_5x_6 + x_6x_7 = 0 \tag{3.5}$$

$$x_1x_3 + x_2x_4 + x_3x_5 + x_4x_6 + x_5x_7 = 0 \tag{3.6}$$

$$x_1x_4 + x_2x_5 + x_3x_6 + x_4x_7 = 0 \tag{3.7}$$

$$x_1x_5 + x_2x_6 + x_3x_7 = 0 \tag{3.8}$$

$$x_1x_6 + x_2x_7 = 0 \tag{3.9}$$

$$x_1x_7 = -1 \tag{3.10}$$

Starting with (3.10), since the elements are to be chosen from the given set, x_1 and x_7 can both be ±1. Without loss of generality, starting with $x_1 = 1$ and $x_7 = -1$ and substituting in equation 3.9 we get

$$x_6 - x_2 = 0.$$

Hence both x_6 and x_2 can take 0, ±1, ±2 values. We can put these values in a tabular form as shown in table 3.1

TABLE 3.1

Sl. no.	x_1	x_2	x_3	x_4	x_5	x_6	x_7
1	1	0				0	-1
2	1	1				1	-1
3	1	-1				-1	-1
4	1	2				2	-1
5	1	-2				-2	-1

We now have 5 possible sets of values of x_1, x_2, x_6, x_7 These can be substituted in equation (3.8) to get

$$x_5 - x_3 = 0 \tag{3.11}$$

$$x_5 - x_3 = +1 \tag{3.12}$$

$$x_5 - x_3 = +1 \tag{3.13}$$

$$x_5 - x_3 = 4 \tag{3.14}$$

$$x_5 - x_3 = 4 \tag{3.15}$$

Equation (3.13) is a repetition of Eq. (3.12) and so also is Eq. (3.15) for Eq. (3.14). Omitting Eq. (3.13) and Eq. (3.15) and solving the rest we have possible solution of

Equation (3.11) as $x_5 = x_3 = 0, \pm 1$, or ± 2,

Equation (3.12) as $x_5 = -2$, $x_3 = 1$ and

as $x_5 = -1$, $x_3 = 0$,

Equation (3.14) as $x_5 = -2$, $x_3 = 2$

Table 3.1 can now be extended to table 3.2

TABLE 3.2

Sl. no.	x_1	x_2	x_3	x_4	x_5	x_6	x_7	REMARKS
1	1	0	0		0	0	-1	SOLUTIONS OF Eqn.(3.11)
2	1	0	1		1	0	-1	
3	1	0	-1		-1	0	-1	
4	1	0	2		2	0	-1	
5	1	0	-2		-2	0	-1	
6	1	1	1		-2	1	-1	SOLUTIONS OF Eqn.(3.12)
7	1	1	0		-1	1	-1	
8	1	2	2		-2	2	-1	SOLUTION of Eqn.(3.14)

By now we have solved all unknowns except x_4 and have 8 sets of solutions shown in table 3.2. Substituting these sets in equation (3.7), we can solve for x_4. In this particular case we find that x_4 can take any of the 5 values $\{0, \pm 1, \pm 2\}$. Hence 8 solutions of table 3.2 will produce 40 solutions.

There are no more unknowns left as we know all possible values of $\{x_1, x_2 \ldots x_7\}$ by now. However, equations (3.4), (3.5), (3.6) are yet to be satisfied. In determining these 40 solutions of $\{x_1, x_2 \ldots x_7\}$, we have taken into account all possible values of specified element set. Hence if at all there is any perfect solution, it must be from these 40.

We substitute all these 40 values in Eqns (3.4), (3.5), (3.6) and check which solution meets the specifications. For E = 18, the only set which satisfies these equations is

$$\{x_1, x_2 \ldots \quad x_7\} = \{1, 2, 2, 0, -2, 2, -1\}$$

We can now generalise the above concept. Starting from equation (3.10) in the foregoing example, we could fix all possible values of various coeffients by the time we reached Eqn.(3.7). Remaining equations (3.6), (3.5) & (3.4) had to be satisfied from the values of x_i's obtained thus far. In other words, we can find all possible sets of various coefficients by solving N/2 equations. Hence N/2 values of specified autocorrelation $R(K), K > N/2$ can be forced. The remaining values are to be checked by actually finding out the autocorrelation with various sets of x_i's found so far.

Having conducted an exhaustive solution of Eqn (3.3), we can positively claim about the existence or otherwise of a sequence matching the specified autocorrelation.

PROGRAMMING ON COMPUTER

The above technique can be implemented on a digital computer. The main part of the implementation consists of solving N/2 equations. Each equation is to be solved with variables taking values from the specified set of integers.

The equations could be solved directly on digital computer. However, for ease of programming on the computer, each equation was first converted into a pseudoboolean equation and then solved. The technique will become clear from the example.

PROBLEM

Find sequences of length 5 satisfying autocorrelation $\{E, 0, 0, 0, 1\}$ with $\{0, \pm 1, \pm 2\}$ as elements.

SOLUTION

Let the required sequence be $(x_1, x_2, x_3, x_4, x_5)$. Elements are x_i's satisfying the constraint $-2 \leq x_i \leq 2$.

Let

$$z_i = x_i + 2$$

therefore $x_i = z_i - 2$.

Hence $\quad -2 \leq x_i \leq 2$ is equivalent to $0 \leq z_i \leq 4$

Now the given autocorrelation can be represented by following set of equations

$$\sum_{i=1}^{N} x_i^2 = E \tag{3.16}$$

$$x_1 x_2 + x_2 x_3 + x_3 x_4 + x_4 x_5 = 0 \tag{3.17}$$

$$x_1 x_3 + x_2 x_4 + x_3 x_5 = 0 \tag{3.18}$$

$$x_1x_4 + x_2x_5 \quad =0 \qquad (3.19)$$

$$x_1x_5 \quad =1 \qquad (3.20)$$

z_i now is an integer lying between 0 & 4. It can be represented in binary form using 3 bits. Therefore

$$z_i = 2^0 \; y_{3i-2} + 2^1 y_{3i-1} + 2^2 y_{3i},$$

where y_i's are boolean variables. Hence

$$x_1 = (z_1 - 2) = y_1 + 2y_2 + 4y_3 - 2 \qquad (3.21)$$

$$x_2 = (z_2 - 2) = y_4 + 2y_5 + 4y_6 - 2 \qquad (3.22)$$

$$\cdot \qquad \cdot \qquad \cdot$$

$$\cdot \qquad \cdot \qquad \cdot$$

$$x_5 = (z_5 - 2) = y_{13} + 2y_{14} + 4y_5 - 2. \qquad (3.23)$$

Starting with the initial values of $x_1 = 1$ & $x_5 = 1$, we have, from (3.21)

$$y_1 = 1, \; y_2 = 1, \quad y_3 = 0 \qquad (3.24)$$

and from Eqn. (3.23)

$$y_{13} = 1, \; y_{14} = 1, \; y_{15} = 0. \qquad (3.25)$$

We can now write (3.19) as

$$(y_1+2y_2+4y_3-2)(y_{10}+2y_{11}+4y_{12}-2)+(y_4+2y_5+4y_6-2)(y_{13}+2y_{14}+4y_{15}-2) = 0.$$

Substituting from equation (3.24) and (3.25) we get

$$4y_{12} + 4y_6 + 2y_{11} + 2y_5 + y_{10} + y_4 = 4 \qquad (3.26)$$

Eqn.(3.26) is in the form of a pseudoboolean equation. All possible solutions of this equation are obtained using method described next.

SOLUTION OF PSEUDOBOOLEAN EQUATION[6]

$$a_1 y_1 + b_1 \bar{y}_1 + a_2 y_2 + b_2 \bar{y}_2 + \cdots a_n y_n + b_n \bar{y}_n = K \quad (3.26)$$

Let the above be the general form of a psuedoboolean equation. We assume that $a_i \neq b_i$ [if not then $(a_i y_i + b_i \bar{y}_i) = a_i$]. First of all we eliminate $\bar{y}_i$ from 3.26 by making a transformation.

$$x_i = y_i \text{ if } a_i > b_i \qquad (3.27)$$
$$\quad\;\; \bar{y}_i \text{ if } a_i < b_i$$

With this we may write

$$a_i y_i + b_i \bar{y}_i = (a_i - b_i)\, x_i + b_i \quad \text{if } a_i > b_i \qquad (3.28)$$

$$= (b_i - a_i)\, x_i + a_i \quad \text{if } a_i < b_i.$$

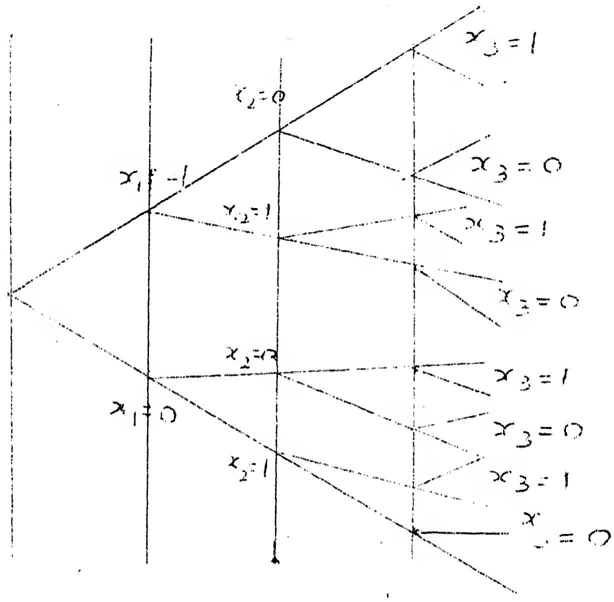
The equation (3.26) becomes

$$c_1 x_1 + c_2 x_2 + \cdots \quad c_n x_n = d \qquad (3.29)$$

where, c_i's ($i = 1, \ldots n$) are constants. Also we assume that we have reindexed c_i's such that $c_1 \geq c_2 \geq \cdots \; c_n > 0$.

We now have to solve equation (3.29) in which all c_i's are > 0. Equation (3.29) can be solved by assigning values to each of the boolean variable x_i. Starting with x_1, it may have two values namely $x_1 = 0$ or $x_1 = 1$. With these

substitutions we change the RHS, and proceed with the new equation with $x_2=0$ and $x_2=1$. This procedure is continued till all solutions are obtained. We can summerise the solution by considering the branches of the tree in Fig.3.1 The tree has n+1 levels 0, 1, ... n.



Tree Showing Solutions of a Psuedoboolean Eqn.

Fig.3.1

Each level r contains 2^r nodes. Each node of the r^{th} level is characterised by the fact that the values of variables $x_1 \ldots \; x_r$ are fixed $(x_1 = n_1, \ldots, x_r = n_r)$, while variables $x_{r+1} \ldots x_n$ are subject to the condition

$$\sum_{j=r+1}^{n} c_j \, x_j = d' \qquad (3.29(a))$$

where $d' = d - \sum_{K=1}^{r} c_k n_k$.

Equation 3.29 (a) is of same type as that of 3.29.

Apparently it looks as if we are going to all the 2^n paths. Fortunately most of them can be avoided by a systematic use of table 3.4.

TABLE 3.4[6]

No.	Case	Conclusions
1	$d < 0$	No solutions
2	$d = 0$	The unique solution is $x_1 = x_2 = \ldots = x_n = 0$
3	$d > 0$ and $c_1 = \ldots \geq c_p > d \geq c_{p+1} \geq \ldots \geq c_n$	The solutions (if any) satisfy $x_1 = \ldots = x_p = 0$ and $\sum_{j=P+1}^{n} c_j x_j = d$
4	$d > 0$ and $c_1 = \ldots c_p = d > c_{p+1} \geq \ldots \geq c_n$	(a) For every $K = 1, 2 \ldots p$: $x_k = 1$ $x_1 = \ldots = x_{K-1} = x_{k+1} = \ldots = x_n = 0$ is a solution. (b) The other solution (if any satisfy, $x_1 = \ldots x_p = 0$ and $\sum_{j=p+1}^{n} c_j x_j = d$

TABLE 3.4[6] (continued)

No.	Case	Conclusions
5	$d > 0$, $c_i < d$ $(i=1\ldots n)$ and $\sum_{i=1}^{n} c_i < d$	No solutions
6	$d > 0$, $c_i < d$ $(i=1\ldots n)$ and $\sum_{i=1}^{n} c_i = d$	The unique solution is $x_1 = x_2 = \ldots\ldots = x_n = 1$
7	$d > 0$, $c_i < d$ $(i=1\ldots n)$ $\sum_{i=1}^{n} c_i > d$ and $\sum_{j=2}^{n} c_i < d$	The solutions (if any) satisfy $x_1 = 1$ and $\sum_{j=2}^{n} c_j x_j = d - c_1$
8	$d > 0$, $c_i < d$ $(i=1,2\ldots n)$; $\sum_{i=1}^{n} c_i > d$ and $\sum_{j=2}^{n} c_j \geq d$	The solution (if any) satisfy either, $x_1 = 1$ and $\sum_{j=2}^{n} c_j x_j = d - c_1$ or, $x_1 = 0$ and $\sum_{j=2}^{n} c_j x_j = d$

Table 3.4 discusses 8 mutually exclusive cases covering all possibilities of solutions of 3.29. Following possibilities may occur.

(i) Equation 3.29 is inconsistent (case 1 & 5)

(ii) Equation 3.29 has a unique solution

(iii) Equation 3.29 is replaced by equation 3.29 (a) (case 3, 4, 7)

(iv) Equation 3.29 is replaced by two equations of type 3.29(a) (case 8)

For case (i) and case (ii) we can exit immediately, but for (iii) & (iv) we have to continue till we exhaust all possibilities.

The above discussed procedure leads to all the solutions of the canonical equation (3.29) If T is the transormation from (3.26) to (3.29), then the solutions of (3.26) are obtained by applying T^{-1} to the solutions of (3.29)

Using this technique the solutions of (3.19) are obtained as shown in Table 3.5

TABLE 3.5

Sl. no.	y_{12}	y_6	y_{11}	y_5	y_{10}	y_4
1	1	0	0	0	0	0
2	0	1	0	0	0	0
3	0	0	1	1	0	0
4	0	0	1	0	1	1
5	0	0	0	1	1	1

We thus have 5 solutions which are to be tried in equation 3.18.

Finally we obtain all possible solutions as given in table 3.6

TABLE 3.6

Sl. no.	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9	y_{10}	y_{11}	y_{12}	y_{13}	y_{14}	y_{15}
1	1	1	0	0	0	0	0	0	1	0	0	1	1	1	0
2	1	1	0	0	0	1	0	0	1	0	0	0	1	1	0
3	1	1	0	0	1	0	0	1	0	0	1	0	1	1	0
4	1	1	0	0	0	1	1	1	0	1	1	0	1	1	0
5	1	1	0	0	1	1	1	1	0	1	0	0	1	1	0

By now all possible values of unknowns are determined. We substitute all these sets into equation 3.17 and after transformation find that the only solutions which satisfy the given autocorrelation for E=14 are

1-2, 2, 2, 1 , 1, 2, 2, -2, 1.

GENERALISATION

We are now in a position to present the method in a systematic and general form.

From Eqn.(3.3) $R(\tau) = \sum_{j=1}^{N-\tau} x_j \, x_{j+\tau} \qquad \tau = 0, 1 \ldots N-1$

where $-p \leq x_i \leq p$ p being an integer.

or $0 \leq z_i \leq 2p$ where $z_i = x_i + p$

Let K be the number of bits required to represent 2p. Then

$$x_i = \sum_{i=0}^{k-1} 2^i \, (y_{i+1})$$

Substituting x_i's in terms of z_i's in Eqn(3.3)

$$R(\tau) = -\sum_{j=1}^{N-\tau} (z_j - p)\ (z_{j+} - p)$$

$$= \sum_{j=1}^{N-\tau} \sum_{i=0}^{K-1} (2^i y_{(j-1)K+1+i} - p) \sum_{i=0}^{K-1} (2^i y_{(j-1)K+1+i+K} - p)$$

$$= \sum_{J=1}^{N-\tau} \left[\sum_{i=0}^{K-1} (2^i y_{q_1} - p) \sum_{i=0}^{K-1} (2^i y_{q_2} - p) \right]$$

$$= \sum_{j=1}^{N-\tau} \left[\sum_{i=0}^{K-1} 2^i\ y_{q_1} \sum_{i=0}^{K-1} 2^i\ y_{q_2} - p \left(\sum_{i=0}^{K-1} 2^i y_{q_1} + \sum_{i=0}^{K-1} 2^i y_{q_2} \right) \right] + (N-\tau)p^2$$

$$\text{or}\quad R(\tau) - (N-\tau)p^2 = \left[\sum_{i=0}^{K-1} 2^i y_{q_1} \sum_{i=0}^{K-1} 2^i y_{q_2} - p \sum_{i=0}^{K-1} 2^i (y_{q_1} + y_{q_2}) \right]_{j=1}$$

(a)

$$\left[\sum_{k=0}^{K-i} 2^i y_{q_1} \sum_{i=0}^{K-1} 2^i y_{q_2} - p\sum_{i=0}^{K-1} 2^i (y_{q_1} + y_{q_2})\right]_{j=N-\tau}$$

(b)

$$+ \sum_{j=2}^{N-\tau-1} \left[\sum_{i=0}^{K-1} 2^i y_{q_1} \sum_{i=0}^{K-1} 2^i y_{q_2} - p \sum_{i=0}^{K-1} 2^i (y_{q_1} + y_{q_2})\right].$$

(c)

by interchanging terms

$$\underset{(LHS)}{a+b} = \underset{(RHS)}{R(\tau) - (N-\tau)\, p^2 - c}$$

For various values of LHS gives the coefficients of pseudoboolean equations. The RHS contains the already known information every time a boolean equation is to be solved.

3.3. SYNTHESIS OF DIFFERENT TYPES OF SEQUENCES.

Given an autocorrelation, and a set of elements, we can thus check whether it is possible to meet these specification or not. Further-more if we are interested in finding an exhaustive listing of a particular class e.g. 'Ternary Barker Sequences' it can be done by generating sequences against an exhaustive list of specified autocorrelation of that class. For example, say we want to find an exhaustive list of Ternary Barker sequences of N=4. The possible autocorrelations are

E, 0, 0, 1	E, 0, 0, -1
E, 0, 1, 1	E, 0, 1, -1
E, 0, -1, 1	E, 0, -1, -1

In the above list all possible values of ending N/2 values of autocorrelations have been taken into account. If we check all these autocorrelations for generating sequences, we will get an exhaustive list of Ternary Barker Sequences of length 4. In rest of this Section we use this idea to generate some other classes of integer sequences.

3.31 LISTING OF VARIOUS TYPES OF INTEGER SEQUENCES

Using techniques described in section 3.2 & 3.3, a number of different types of sequences as listed below, have been generated.

(a) INTEGER SEQUENCES WITH $|C_0| = |C_N| = \pm 1$

- (i) Barker Sequences
- (ii) Ternary Barker Sequences upto length 15
- (iii) Quinquinary Broad Barker Sequences up to length 13
- (iv) Quinquinary Barker Sequences up to length 13
- (v) Quinquinary Integer Huffman Sequences up to length 32
- (vi) Quinquinary Broad Huffman Sequences up to length 32

(b) Integer Sequences with $|C_0| = |C_N| = 1$ and elements $\{0, \pm 1, \ldots \pm 7\}$ up to length 8.

(c) Integer Sequences with $|C_0| \neq |C_N|$ and elements $0, \pm 1, \ldots \pm 7\}$ up to length 8

(a) (i) BARKER SEQUENCES have already been defined and are listed extensively in literature. The elements of Barker sequences a re restricted to ± 1

(ii) TERNARY BARKER SEQUENCES

Sequences such that $|R(K)| \leq 1, K \neq 0$ are called[4] Ternary Barker Sequences, if the elements are not restricted to ± 1. Some of the Ternary Barker Sequences up to length 10 have been listed by[4]. Using the techniques of Section 3.3, Ternary Barker Sequences up to length 15 are exhaustively listed in Appendix 'A'

(iii) QUINQUINARY BARKER SEQUENCES

Sequences such that $|R(K)| \leq 1, K \neq 0$ will be called Quinquinary Barker Sequences if the elements are allowed to be from $\{0, \pm 1, \pm 2\}$. An exhaustive listing of Quinquinary Barker Sequences up to length 13 is placed at Appendix 'B'.

(iv) QUINQUINARY BROAD BARKER SEQUENCES

Barker Sequences are sequences with ± 1 as elements such that $|R(K \neq 0)| \leq 1$. As an extension of the concept, the sequences such that $R(K > K_0 > 1) \leq 1$ but elements $(0, \pm 1, \pm 2)$ will be called Quinquinary Broad Barker sequences. We would normally be

interested in the smallest value of K_0. As examples, the sequence (1, 1, -1, -1, 1, -1, 1, -1) has R(K) = (8, -3, 0, -1, 0, 1, 0, -1) and is a Broad Barker Sequence of length 8 with K_0 =1. An exhaustive list of Broad Barker sequences up to length 13 is placed at Appendix 'C'.

(v) QUINQUINARY INTEGER HUFFMAN SEQUENCES

Sequences for which $|R(K)| = 0 \quad K \neq 0, N-1$ and elements are restricted to $\{0, \pm 1, \pm 2\}$ will be called Quinquinary Integer Huffman sequences. The only sequences obtained are (1, 2, 2, -2, 1) and (1, 2, 2, 0, -2, 2, -1). After an exhaustive search it is found that no such sequences exist up to length 32.

(vi) QUINQUINARY BROAD HUFFMAN SEQUENCES

As an extension of the concept of (v), sequences such that $R(K \sim K_0 > 1) = 0 \quad K \neq (0, N-1)$ will be called Quinquinary Broad Huffman Sequences. A list of some of these sequences is given in Appendix 'D'.

(b) INTEGER SEQUENCES WITH $|C_0| = |C_N|$ and elements $\{0, \pm 1, \pm 2, \ldots \pm 7\}$.

We have seen that sidelobe ratio of sequences considered so far is limited since the element size has been restricted

to $(0, \pm1, \pm2)$. In order to get higher central to sidelobe ratios the element range was increased to $\{0, \pm1, \ldots \pm7\}$ An exhaustive list of such sequences is placed at Appendix 'E'.

(c) INTEGER SEQUENCES WITH $|C_0| \neq |C_N|$ and elements $\{0, \pm1, \pm2 \ldots \pm7\}$

By relaxing the condition on $|C_0|$ and $|C_N|$, it is possible to obtain more sequences with higher sidelobe ratios. Some of these sequences are listed in Appendix 'F'.

Computer programmes for synthesizing 'a' is given in Appendix 'K' while for 'b' & 'c' is given in Appendix 'L'.

CHAPTER 4

SYSTEM IDENTIFICATION OF LINEAR SYSTEMS

The conventional method of determining empirically the dynamic characteristics of a linear system (or part of it) is by means of either transient or sinusoidal inputs. Although for linear systems, the two methods yield equivalent information, the use of step input in practice tends to give rise to saturation effects, if the magnitude of the step is well above the system noise level. The use of sinusoidal inputs is not usually so limited by saturation effects. However, the method is time consuming since steady state-measurements have to be taken at many different frequencies. Furthermore the test signal must again be well above the noise level.

In measuring the characteristics of some systems, either it is desirable to disturb the system as little as possible or a rapid automated method must be employed. If only small test disturbances to the system can be tolerated, the total time must be long. Conversly if test inputs that are well above the system noise are permissible, a rapid determination is possible.

The principle of the method hinges on the well known theoretical result, that if white noise is applied to a linear system, the crosscorrelation of input and output

gives the system impulse response. Let the input be $x(t)$ and the output be $y(t)$, then the system impulse response at time $t = \tau$ is given by

$$h(\tau) = \lim_{T \to \infty} \frac{1}{2T} \int_{-T}^{T} x(t)\, y(t+\tau)\, dt \qquad (4.1)$$

However, the practical exploitation of this result leads to some interesting problems. First white noise (characterised by a flat power spectrum of infinite bandwidth) is a theoretical concept, it is awkward to generate a flat power spectrum at the low frequencies and difficult to achieve repeatable results. Secondly the integration in equation (4.1) must be over a finite interval which in some applications like data communication channels, must be as short as possible. Because of these considerations a true stochastic input cannot be employed, instead a ramdom input which repeats itself with period T can be employed. Such an input can in principle be proportioned to approximate very closely to the desired test input and the integration is carried over a finite interval qT, where q is an integer. A PN sequence could be easily used as a test signal. However while using PN sequence, it is necessary that the sequence be applied to the system, for at least one period before correlation commences, in order to ensure that initial transients due to the application of the sequence to the system have disappeared. Because of this requirement, the

minimum identification time, in the absence of noise, is in the region of 3 code lengths.

This estimation time could be reduced, if a test signal with the following properties were available.

a) For zero shifts, the autocorrelation function (ACF) should approximate to an impulse.

b) For other shifts, the (ACF) should be small (Nearly Zero) ensuring that entire ACF is a reasonable approximation to white noise.

c) The signal should be of finite length. This property eliminates the requirement that the system should settle to a steady state value by removing the periodicity of the test signal. This in turn leads to an ACF of finite length.

d) The sequence should be reproduci-ble.

USE OF BARKER SEQUENCES[8]

A suitable set of sequences satisfying above properties are Barker sequences. It has, however, been found that at least two code lengths are required for Identification. Also due to short code lengths and consequently limited central to side lobe ratio, Barker codes are not ideally suited for System Identification.

USE OF INTEGER HUFFMAN SEQUENCES

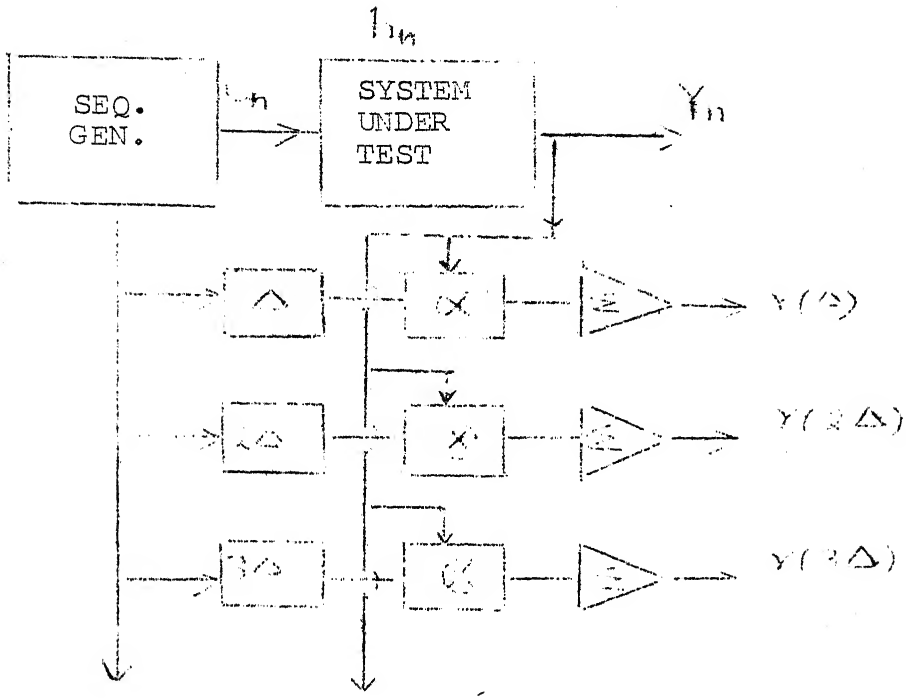
Huffman sequences satisfy all the properties as enumerated earlier. In addition as the ACF is zero except for 0 & N-1 shifts, only one sequence length is required for System Identification. Integer sequences are particularly suitable for this purpose since they can be easily generated and transmitted over digital networks. In this study, System Identification, using the three methods, namely PN Sequence, Barker sequence and Huffman sequence has been compared by simulating on a digital computer.

4.1 SYMBOLS

C_K	Input sequence
a_i	Amplitude of i^{th} member of sequence
h_K	Impulse response of system under test
$r(\tau)$	Out put of the summing CCT for code shift of τ.
$R(\tau)$	Autocorrelation function of C_K at delay τ.
Δ	Bit interval of various sequences
N_1	Length of sequence
N_2	Length of Impulse response sequence
N	$(N_1 + N_2 - 1)$
Δ	Delay of one unit.

4.2 IDENTIFICATION USING PSUEDORANDOM SEQUENCES[5]

Consider the cross-correlation scheme shown in Fig.4.1



CROSS-CORRELATION SCHEME USING VARIOUS SEQUENCES

Fig. 4.1

A test input C_n is applied to a system whose impulse response is h_n . The output of the system is y_n . The output y_n of the system is correlated with the delayed versions of test signal through a multiplier and a summing circuit. The output of correlator is denoted

$r(\tau)$, $\tau = \Delta, 2\Delta, \ldots$ where Δ represents a unit delay.

Now

$$r(\tau) = \sum_{1}^{N} y_n \, C_{n-\tau+1}$$

where

$$y_n = \sum_{k=1}^{n} C_{n-k+1} \, h_k$$

$$n = 1 \,..\, N, \qquad K = 1 \,..\, n, \qquad N = N_1 + N_2 - 1.$$

Substituting y_n:

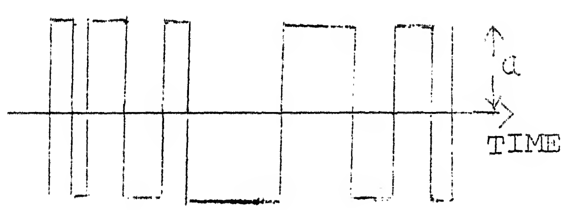
$$r(\tau) = \sum_{1}^{N} \sum_{1}^{n} C_{n-K+1} C_{n-\tau+1} h_K$$

$$r(\tau) = \sum_{1}^{n} R(K-\tau) h_K \qquad (4.2)$$

where

$$R(K-\tau) = \sum_{n=1}^{N} C_{n-K+1} \, C_{n-\tau+1}$$

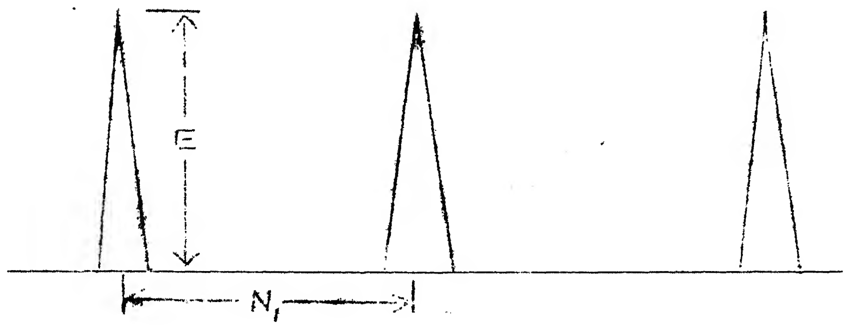
Now consider a periodic PN sequence C_n as shown in Fig. 4.2



PERIODIC PN SEQUENCE

Fig. 4.2

Its periodic autocorrelation can be approximated to Fig. 4.3



APPROXIMATE AUTOCORRELATION OF A PN SEQUENCE.

Fig. 4.3

$$R(K-\tau) = E \sum_{M=0}^{\infty} \delta(K-\tau-MN_1)$$

substituting into Equation 4.2

$$R(\tau) = E \sum_{1}^{n} h_K \sum_{M=0}^{\infty} \delta(K-\tau-MN_1)$$

$$= E\left[h(\tau) + h(\tau+N_1) + h(\tau+2N_1)+\ldots\right]$$

Assuming the system impulse response to be negligible after N_1 we have

$$r(\tau) = E\,h(\tau)$$

or $$h(\tau) = r(\tau)/E$$

$h(\tau)$ can thus be calculated from a knowledge of $r(\tau)$

4.3 IDENTIFICATION USING BARKER SEQUENCES[8]

Consider again the cross-correlation scheme as shown earlier in Fig. 4.1. Let C_n now be a Barker sequence. Once again

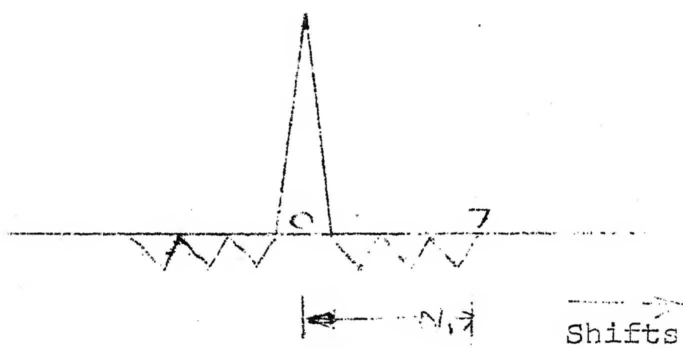
$$r(\tau) = \sum_{1}^{N} \sum_{1}^{n} C_{n-K+1}\, C_{n-\tau+1}\, h_K$$

$$= \sum_{1}^{n} R(K-\tau)\, h_K$$

where

$$R(K-\tau) = \sum_{m=1}^{N} C_{m-K+1}\, C_{m-\tau+1} \; .$$

Now consider the form of $R(\tau)$ shown in Fig. 4.4

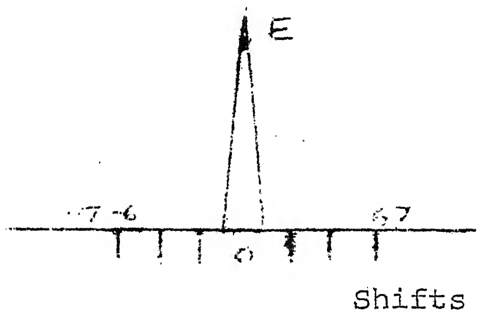


AUTOCORRELATION OF A BARKER SEQUENCE $N_1 = 7$

Fig. 4.4

By means of a suitable approximation for $R(\tau)$ the impulse equivalent nature of $R(\tau)$ may be utilised to solve equation 4.2

A suitable approximation of autocorrelation function of length 7 is shown in Fig. 4.5



APPROXIMATION OF AUTOCORRELATION OF BARKER SEQ. $N_1 = 7$

Fig. 4.5

The approximation can be written as

$$R(K-\tau) = \frac{N_1+1}{N_1} \left[E\,\delta\,(K-\tau) \right] - \frac{E}{2N_1} \left[\left\{ 1+(-1)^{K-\tau} \right\} \right] \quad (4.3)$$

For length 5 or 13 the approximation is

$$R(K-\tau) = \frac{N_1-1}{N_1} \quad E\,\delta\,(K-\tau) - \frac{E}{2N_1} \left\{ 1+ (-1)^{K-\tau} \right\} \quad (4.4)$$

Substituting for $R(K-\tau)$ from eqn. (4.3) to eqn. (4.2)

$$r(\tau) = \sum_{1}^{n} h_K \frac{N_1+1}{N_1} (E)\,\delta\,(K-\tau) + h_K \left(\frac{E}{2N_1}\right) \left\{ 1+(-1)^{K-\tau} \right\}$$

$$\text{or} \quad r(\tau) = \frac{N_1+1}{N_1} Eh(\tau) - \frac{E}{2N_1} \sum_{K=1}^{n} h_K + \sum_{K=1}^{n} h_K (-1)^{K-\tau}$$

$$\text{or} \quad r(\tau) = \frac{N_1+1}{N_1} E\, h(\tau) - \left\{ K_1 + (-1)^{K-\tau} K_1 \right\}$$

Thus the error present in the output of summing circuit is either zero or some other value depending upon the value of τ. To eliminate the error term if the sequence is advanced, rather than delayed prior to multiplication stage, the summing circuit output will be

$$r(-\tau) = 0 - \frac{E}{2N_1} \sum_{K=1}^{n} h_K + \sum_{K=1}^{n} h_K (-1)^{K-\tau}$$

$$= \left\{ - K_1 + (-1)^{K-\tau} K_1 \right\}$$

Thus the required terms consisting of K_1 could be generated.

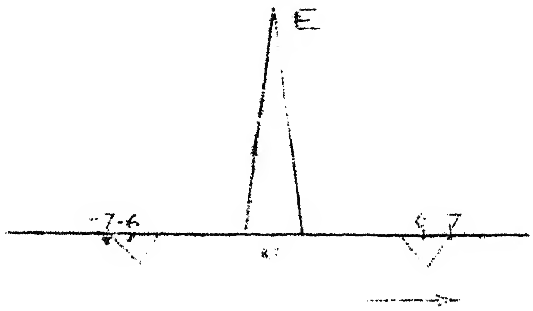
From these results it is apparent that identification using Barker sequences for shifts which are integral multiples of bit interval, requires two corrective correlations (as against with periodic sequences) but on the whole identification is reduced to two lengths as against 3 required by periodic sequences.

4.4 IDENTIFICATION USING INTEGER HUFFMAN SEQUENCES

Consider again the correlative scheme shown in Fig. 4.1 If the input sequence is now Huffman Sequence, once again

$$r\ (\tau) = \sum_{1}^{n} R\,(K-\tau)\ h_K \qquad (4.5)$$

Now consider the form of autocorrelation function of a Huffman sequence shown in Fig. 4.6



AUTOCORRELATION OF HUFFMAN SEQUENCE $N_1 = 7$

SEQ $\{1,\ 2,\ 2,\ 0,\ -2,\ 2,\ -1\}$

Fig. 4.6

We can represent this as

$$R\,(K-\tau) = E\,\delta(K-\tau) + \delta\{K-\tau-(N_1-1)\} + \delta\{K-\tau+(N_1-1)\} \qquad (4.6)$$

Assuming that the system impulse response is zero at shift (N_1-1) and beyond. Substituting Eqn.(4.6) into Eqn (4.5)

$$r\ (\tau) = E \sum_{1}^{n} (K-\tau)\ h_K + 0$$

Therefore

$$r\ (\tau) = E\ h(\tau)$$

or

$$h\ (\tau) = \frac{r(\tau)}{E}$$

Interestingly since the autocorrelation function of a Huffman sequence is ideal impulse equivalent, no errors are involved in the estimation in the first run of the sequence itself. It therefore requires only one length as against two of Barker and 3 of Pseudorandom sequences.

4.5 NOISE PERFORMANCE

When a noise source n_K is present at the output of the system under investigation, an additional error term is present in the estimated response. The output from the summing circuit, for shifts, which are integral multiples of bit interval is now

$$r(\tau) = \sum_{1}^{n} R(K-\tau)\ h_K + \sum_{1}^{N_2} n_K\ C_{K-\tau}$$

It is found by[5] that with Pseudorandom sequences the error can be reduced if the period of summation is increased from N_1 to qN_1 where q is an integer. It is found that Mean Square Error is inversely proportional to square root of q

Similar results are observed even for Barker and Huffman sequences as seen in subsequent sections.

4.6 COMPARISON OF VARIOUS METHODS OF ESTIMATION

All the three correlative schemes discussed, were implemented using a digital computer. The zero mean white Gaussian noise samples $W_n(K)$ with variance ^{2} were generated by using

the Box Muller method. The noise samples were generated by

$$W_N(K) = \sigma \left[-2 \ln (R_1) \right]^{\frac{1}{2}} \cos (2\pi R_2)$$

where R_1 and R_2 are uncorrelated, uniformly distributed random numbers in the range 0 and 1. For linear systems, noise variance is related to SNR through

$$\sigma^2 = 10^{-\frac{1}{SNR}}$$ where SNR is specified in db.

SIMULATION RESULTS

A comparison of performance was done under various conditions of noise. The result obtained by various methods are placed in table 4.1 (shown in next page) Both First and Second order systems were tested. In both cases the maximum and the meansquare error were computed. The results obtained were also plotted graphically and are shown from Fig. 4.7 to 4.12 Computer programmes for all the three methods are given in Appendix M to Appendix O.

Contd....

TABLE 4.1

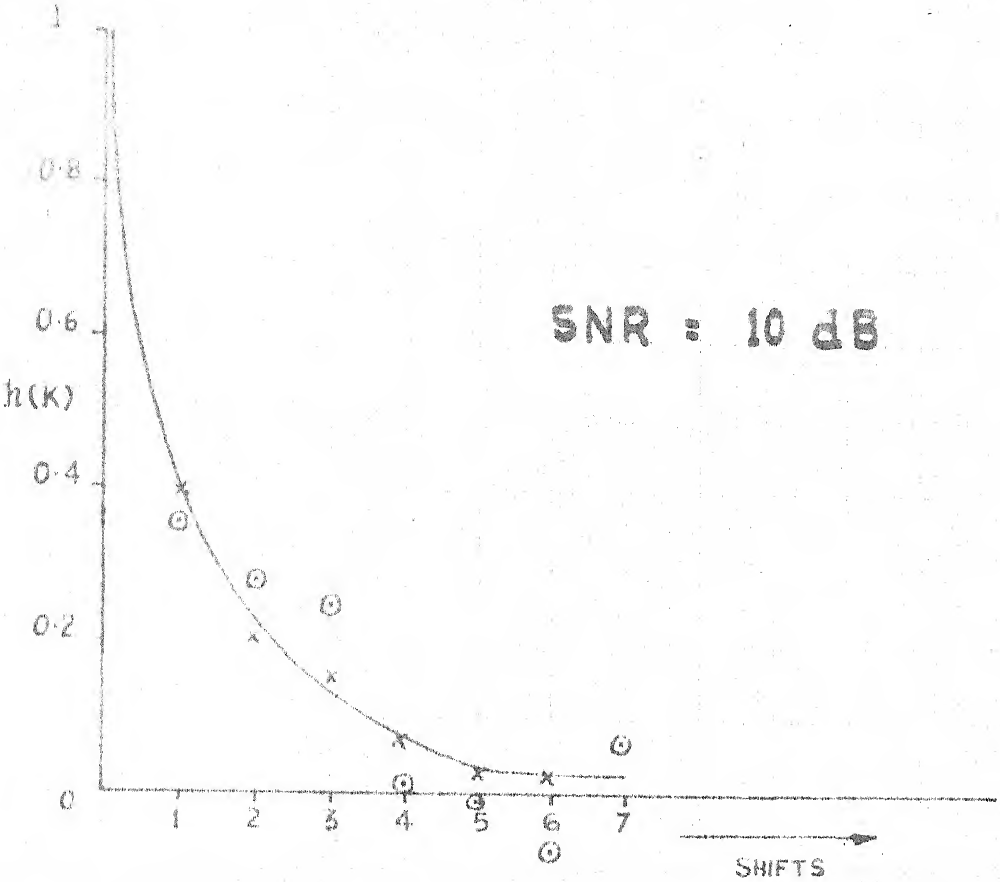
COMPARISON OF CORRELATIVE IMPULES RESPONSE ESTIMATION METHODS

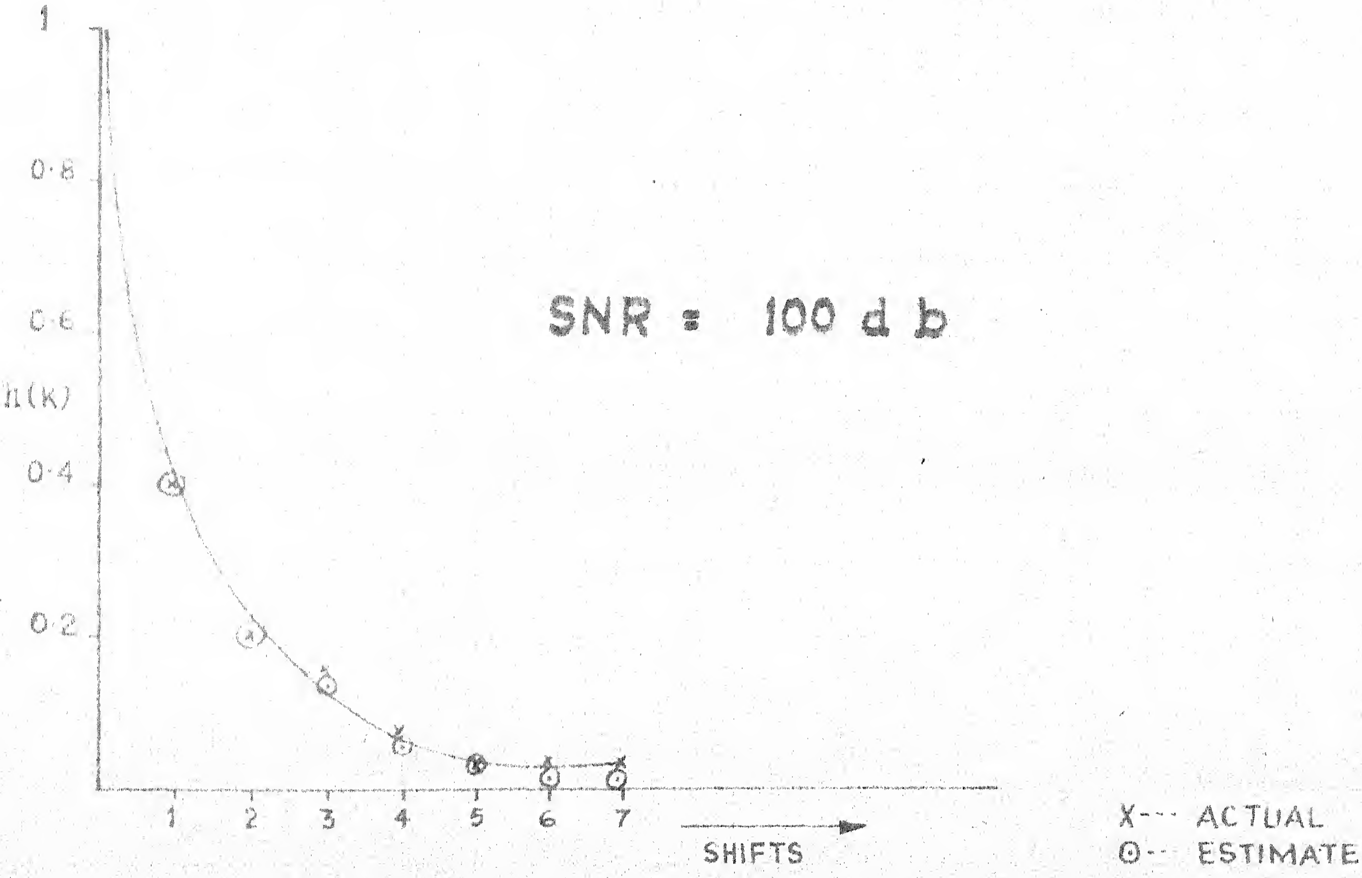
Class of System	No. of Avr.	Type of Input	Signal to Noise Ratio (DB)							
			10		20		50		100	
			Max err.	M.S. Err.	Max Err.	M.S. Err.	Max Err.	M.S. Err.	Max Err.	M.S. Err.
I	1	Barker	0.48	0.20	0.08	0.05	0.00	0.00	0.00	0.00
		PNSEQ.*	0.11	0.04	0.03	0.01	0.01	0.00	0.01	0.00
		HUFFMAN	0.02	0.01	0.01	0.00	0.00	0.00	0.00	0.00
	16	BARKER	0.05	0.05	0.02	0.01	0.00	0.00	0.00	0.00
		PNSEQ.*	0.08	0.00	0.01	0.00	0.01	0.00	0.01	0.00
		HUFFMAN	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00
II	1	BARKER	0.48	0.20	0.08	0.05	0.00	0.00	0.00	0.00
		PNSEQ.*	0.11	0.04	0.03	0.01	0.01	0.00	0.01	0.00
		HUFFMAN	0.02	0.01	0.01	0.00	0.00	0.00	0.00	0.00
	16	BARKER	0.05	0.03	0.02	0.01	0.00	0.00	0.00	0.00
		PNSEQ.*	0.08	0.00	0.01	0.00	0.01	0.00	0.01	0.00
		HUFFMAN	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00

* RESULTS HAVE BEEN TABULATED AFTER CORRECTING FOR BIAS ERROR IN THIS CASE.

4.8 CONCLUSION

From a perusal of results it is observed that while the performance of all methods is equally good under low noise (100 Db) condition, it is not so, under high noise conditions.

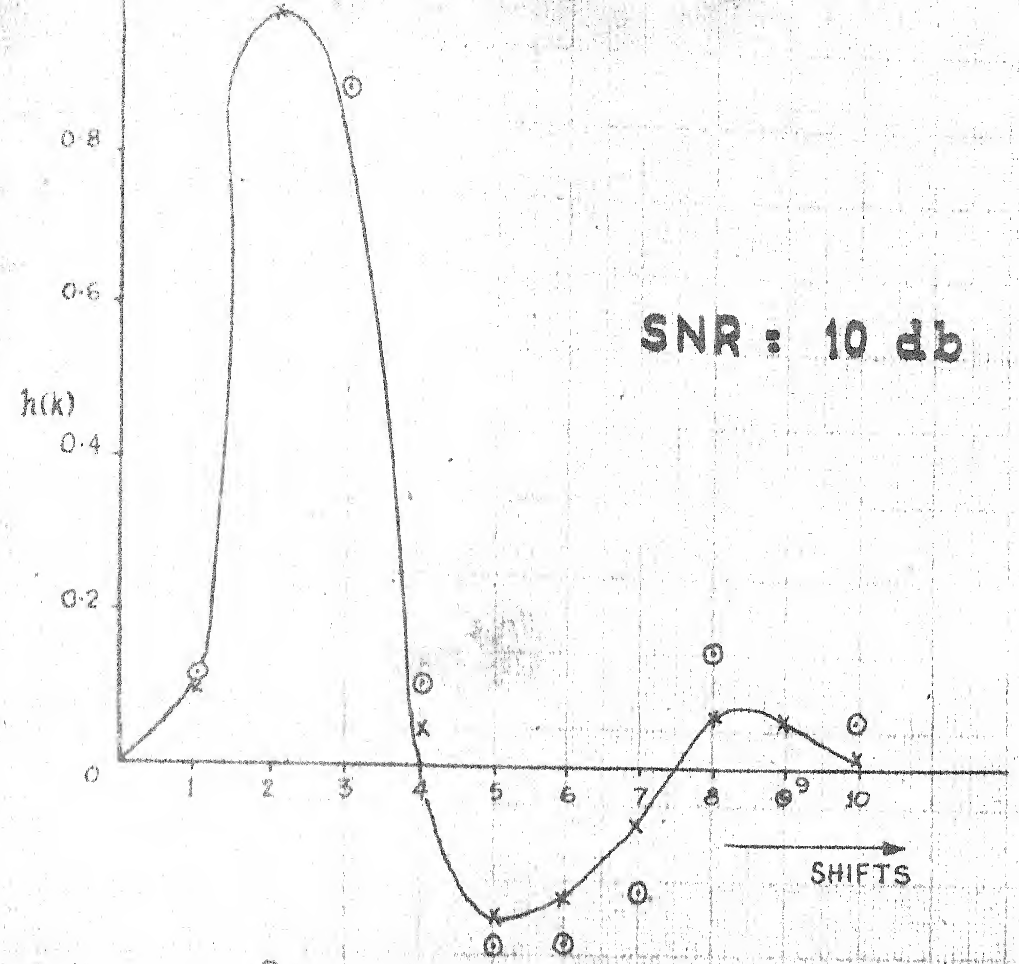


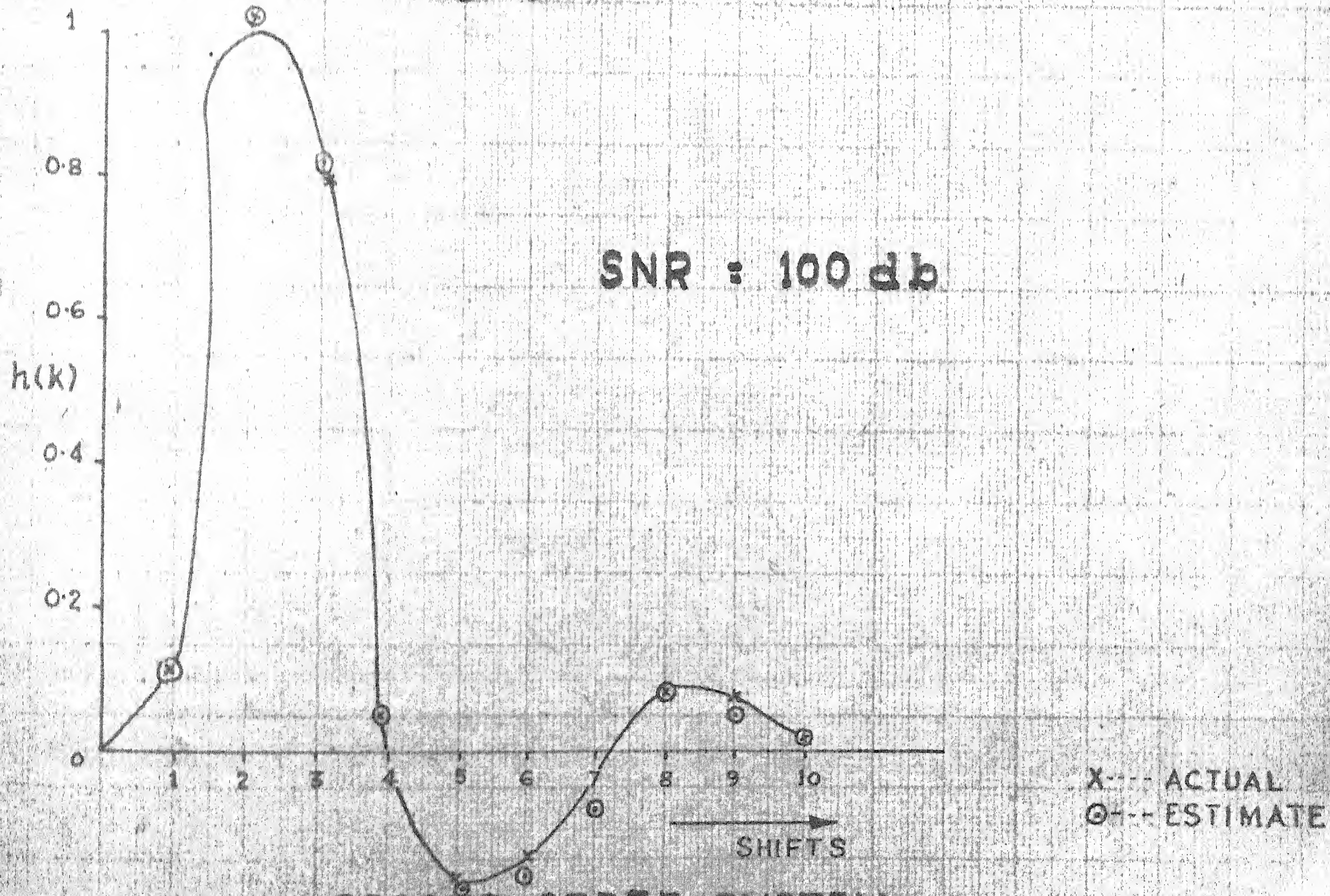


FIRST ORDER SYSTEM

IMPULSE RESPONSE USING PN SEQUENCE L=31

Fig 4.7

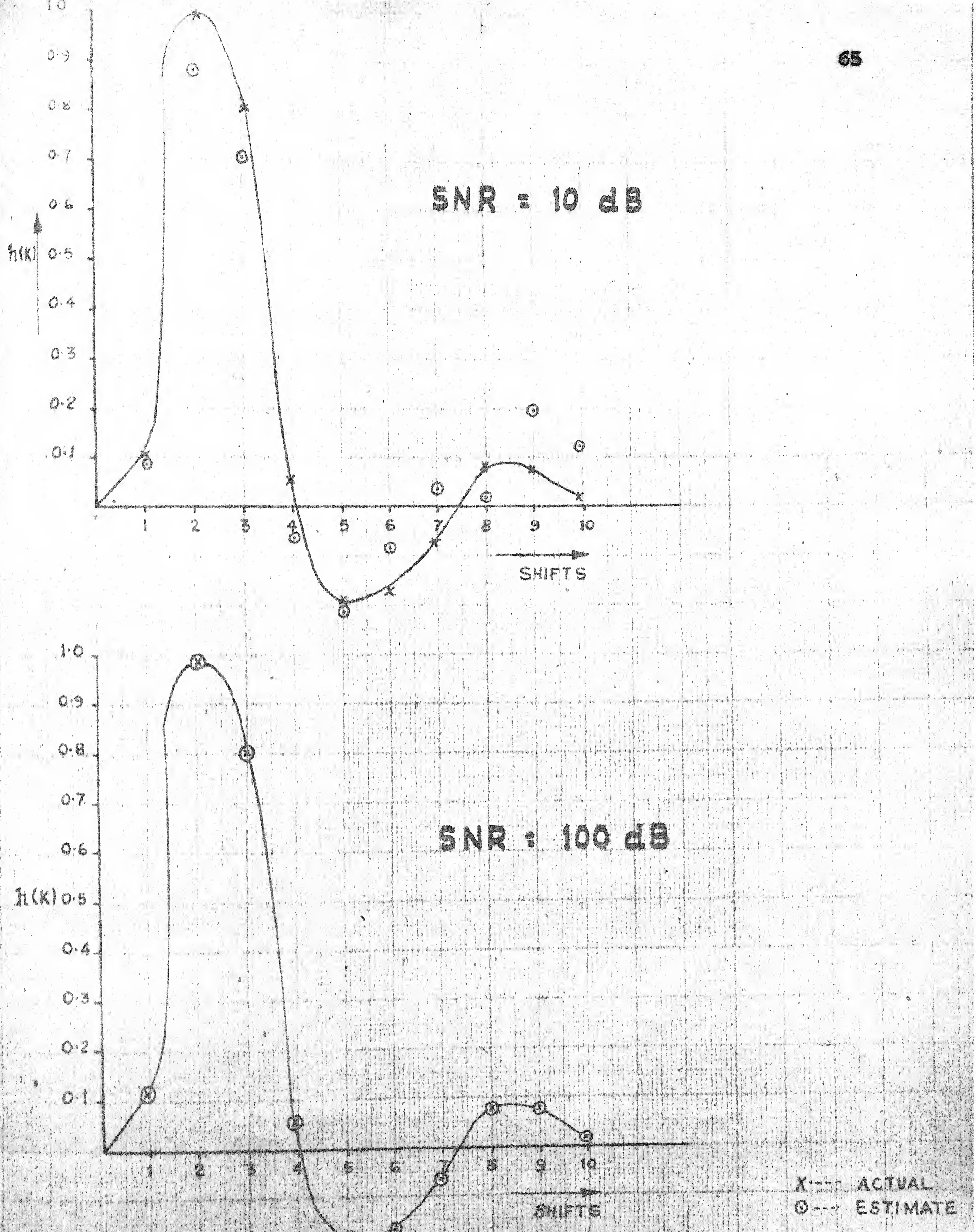




SECOND ORDER SYSTEM

IMPULSE RESPONSE USING PN SEQUENCE L:31

Fig 4·8



SECOND ORDER SYSTEM

IMPULES RESPONSE USING BARKER SEQUENCE L=13

Fig. 4.10

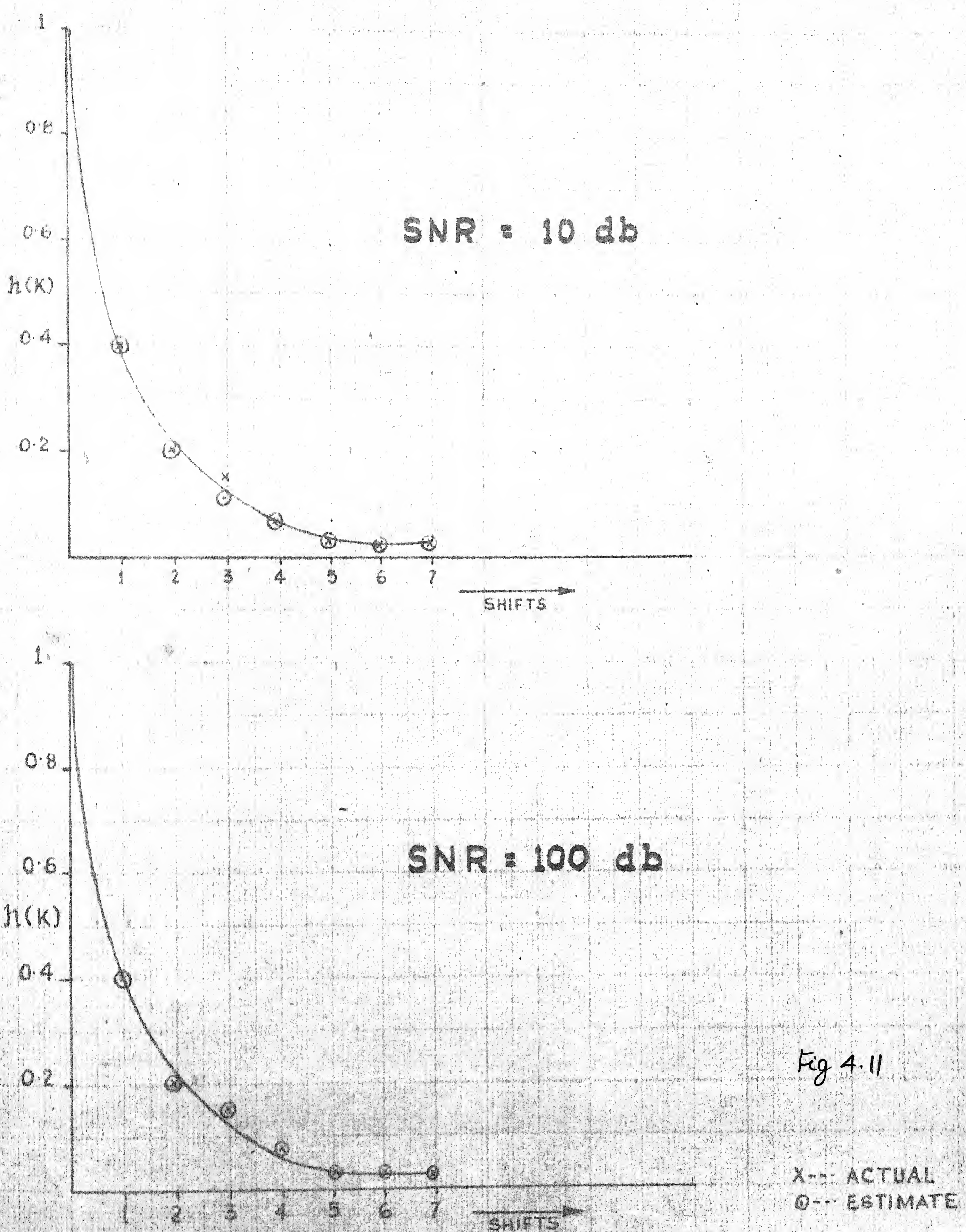


Fig 4.11

FIRST ORDER SYSTEM

IMPULSE RESPONSE USING HUFFMAN SEQUENCE

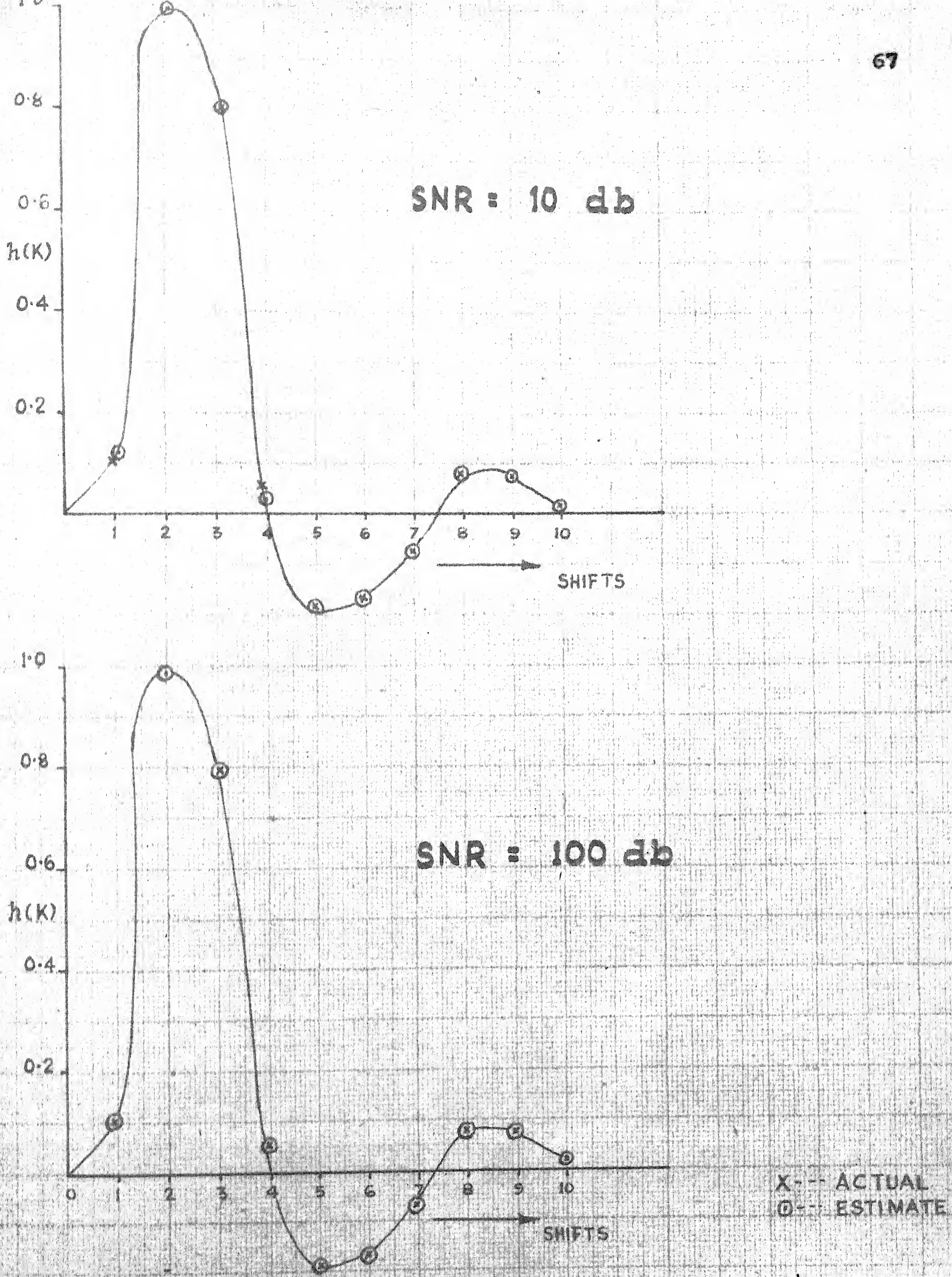


Fig. 4.12

SECOND ORDER SYSTEM

IMPULSE RESPONSE USING HUFFMAN SEQUENCE L = 11

A Huffman sequence is found to be giving the best estimate of impulse response. In addition, against 3 sequence lengths of PN Sequence and 2 of Barker sequence, a Huffman sequence requires only one length for impulse response estimation. For a given system, however, the use of a Huffman sequence is limited by the fact that since the individual elements are large, it may cause saturation in the system.

CHAPTER 5

CONCLUSIONS AND SCOPE FOR FUTURE WORK

From a study of chapter 2 we observe that integer Huffman sequences satisfying $C_0 = C_N = 1$ do not exist for all lengths and that as the length is increased, the efficiency falls rapidly. Consequently longer integer Huffman sequences of above type will be of little use. Subsequently we found another set of integer Huffman sequences where in the length was 2^n, $n = 1, 2 \ldots$ In this case $|C_0| \neq |C_N|$ and here again as the length was increased the efficiency became lower.

In view of the limitations of integer Huffman sequences, we tried to develop a method by which we could synthesize integer sequences with good auto-correlation properties and consisting of elements from a small set eg. $\{0, \pm 1, \pm 2\}$, $\{0, \pm 1, \pm 2 \ldots \pm 7\}$ Here we observed that as the length of the sequence was increased, maximum central to side lobe ratio tended to saturate.

These drawbacks could be overcome to some extent, if we permit the element set to contain powers of $\frac{1}{2}$ as well eg. $0, \pm 1, \pm\frac{1}{2}, \pm 2, \pm\frac{1}{4}$. This will give following advantages.

a) The sequence containing $\pm\frac{1}{2}$, $\pm\frac{1}{4}$ etc. as elements, could still be implemented easily like pure integer sequences.

b) As the Element Spread (The difference between the Maximum and Minimum Element) is not increased, good efficiency could be maintained even for larger sidelobe ratios.

In view of the above, the problem of finding a sequence with good autocorrelation could be redefined as

$$
\begin{aligned}
R(K) &= E \text{ for } K = 0 \\
&= L \text{ for } K = \{1, 2, \ldots N-2\},\ L_1 < L < L_2 \\
&= m \text{ for } K = N-1 \\
&= 0 \text{ for } K > N
\end{aligned}
$$

a) If $L_1 = -1$, $L_2 = +1$, $L = 0, \pm 1$, $m = \pm 1$ we get autocorrelation function of a Barker sequence.

b) If $L_1 = L_2 = L = 0$ and $m = \pm 1$ we get autocorrelation function of a Huffman sequence.

c) If $L_1 = -m$, $L_2 = m$, $m, L = \{1, \pm 1/2, \pm 1/4, \pm 1/8 \ldots\}$ we get an autocorrelation function, which lies in between Barker & Huffman autocorrelations. The sequence matching this autocorrelation would consist of elements $\{0, \pm 1, \pm 1/2, \pm 2, \pm 1/4 \ldots\}$ thus retaining all the advantages of an integer sequence and yet giving better efficiency and sidelobe ratio. This could be further explored.

contd....

In Chapter 4, we studied the feasibility of using integer Huffman sequences for system identification and found that it was advantageous to use these sequences, provided, the system does not get saturated because of large variations in the elements of the sequence. Possibility of using these sequences for synchronisation and other applications needs more investigation.

--

REFERENCES

1 D.A.Huffman- The Generation of Impulse equivalent pulse trains- IRE. Trans. Inf. Theory, Vol.IT-8 No.5, pp S10- S16, 1962.

2 M.H.Ackroyd- Some Integer Huffman Sequences- IEEE Trans. on Inf.Theory, Vol.IT-26, No.1, Jan.80

3 Eli Brookner - Radar Technology- Artech 1978.

4 P.S.Moharir - Terminal Admissibility Techniques for Signal Design- CIP Report No. 43, Indian Institute of Science Bangalore.

5 Hazlerigg and Noton-System Identification using Cross correlating Equipment- Proc.IEE.Vol.112, No.12, Dec.65, pp 2385-95.

6 P.L.Hammer- Boolean Methods in operation Research and Related Areas- Springer Verlag New York.

7 Ray. H.Petit- Pulse sequences with good Autocorrelation Properties- Microwave Journal, Feb. 67.

8 H.Wilson- Barker Sequences in System Identification Techniques- Control Jan.69.

APPENDIX A
TERNARY BARKER SEQUENCES

L	SEQUENCE	AUTOCORRELATION	E	EFF
5	1, -1, 0, 1, 1	4, 0, -1, 0, 1	4	0.8
	1, 1, 0, 1, -1	4, 0, 1, 0, -1	4	0.8
	1, -1, -1, 0, -1	4, 0, 0, 1, -1	4	0.8
	1, 1, -1, 0, -1	4, 0, 0, -1, -1	4	0.8
6	1, -1, 0, 1, 1, 1	5, 1, 0, 0, 0, 1	5	0.83
	1, 0, -1, 1, 1, 1	5, 1, -1, 0, 1, 1	5	0.83
	1, 0, 1, -1, 1, 1	5, -1, 1, 0, 1, 1	5	0.83
	1, 0, 1, -1, -1, 1	5, -1, -1, 0, -1, 1	5	0.83
	1, 1, 0, -1, 1, -1	5, -1, 0, 0, 0, -1	5	0.83
	1, -1, 1, 1, 0, -1	5, -1, -1, 0, 1, -1	5	0.83
	1, -1, -1, -1, 0, -1	5, 1, 1, 0, 1, -1	5	0.83
	1, 1, -1, -1, 0, -1	5, 1, -1, 0, -1, -1	5	0.83
7	1, -1, 0, 0, 1, 1, 1	5, 1, 1, -1, 0, 0, 1	5	0.71
	1, -1, 1, 0, 0, 1, 1	5, -1, 1, 1, 0, 0, 1	5	0.71
	1, 0, 1, 0, -1, -1, 1	5, 0, -1, -1, 0, -1, 1	5	0.71
	1, 0, 1, 0, -1, 1, 1	5, 0, -1, 1, 0, 1, 1	5	0.71
	1, 0, -1, 1, 0, 1, 1	5, 0, 0, 1, -1, 1, 1	5	0.71
	1, 0, -1, -1, 0, -1, 1	5, 0, 0, -1, -1, -1, 1	5	0.71
	1, 0, 1, -1, 0, 1, 1	5, 0, 0, -1, 1, 1, 1	5	0.71
	1, 0, 1, 1, 0, -1, 1	5, 0, 0, 1, 1, -1, 1	5	0.71
	1, 1, 1, -1, 0, 1, -1	6, 0, -1, 1, 0, 0, -1	6	0.86

TERNARY BARKER SEQUENCES (contd)

L	SEQUENCE	AUTOCORRELATION	E	EFF
7	1, 1, 1, 0, 0, 1, -1	5, 1, 1, 1, 0, 0, -1	5	0.71
	1, 1, 0, -1, -1, 1, -1	6, 0, -1, -1, 0, 0, -1	6	0.86
	1, 1, 0, 0, -1, 1, -1	5, -1, 1, -1, 0, 0, -1	5	0.71
	1, -1, 1, 1, 0, -1, -1	6, 0, -1, -1, 0, 0, -1	6	0.86
	1, 1, 1, -1, 1, 0, -1	6, 0, 0, 1, 0, -1, -1	6	0.86
	1, 1, -1, 0, -1, 0, -1	5, 0, 1, -1, 0, -1, -1	5	0.71
	1, 0, 1, 0, 1, -1, -1	5, 0, 1, -1, 0, -1, -1	5	0.71
	1, 0, -1, 1, -1, -1, -1	6, 0, 0, 1, 0, -1, -1	6	0.86
	1, -1, 1, 1, 1, 0, -1	6, 0, 0, -1, 0, 1, -1	6	0.86
	1, -1, -1, 0, -1, 0, -1	5, 0, 1, 1, 0, 1, -1	5	0.71
	1, -1, 1, 1, 0, 0, -1	5, -1, 0, 0, -1, 1, -1	5	0.71
	1, 1, 1, -1, 0, 0, -1	5, 1, 0, 0, -1, -1, -1	5	0.71
	1, 1, 1, 0, -1, 1, -1	6, 0, 1, 0, -1, 0, -1	6	0.86
	1, -1, -1, -1, 0, 0, -1	5, 1, 0, 0, 1, 1, -1	5	0.71
	1, 1, -1, 1, 0, 0, -1	5, -1, 0, 0, 1, -1, -1	5	0.71
8	1, 1, 1, 0, -1, 0, 1, -1	6, 1, -1, 0, 0, 0, 0, -1	6	0.75
	1, 1, -1, -1, 0, -1, 1, -1	7, -1, 0, -1, -1, 1, 0, -1	7	0.88
	1, -1, 1, 0, 1, 1, 0, -1	6, -1, 1, -1, 0, 0, 1, -1	6	0.75
	1, -1, 1, 1, 1, 1, 0, -1	7, 1, 1, 0, -1, 0, 1, -1	7	0.88
	1, -1, 0, -1, -1, 1, 0, -1	6, -1, -1, 1, -1, 1, 1, -1	6	0.75
	1, 0, -1, -1, -1, 0, 1, -1	6, 1, -1, -1, -1, 1, 1, -1	6	0.75
	1, 1, 1, 0, -1, -1, 1, -1	7, 1, 0, -1, -1, -1, 0, -1	7	0.88

TERNARY BARKER SEQUENCES (Contd)

L	SEQUENCE	AUTOCORRELATION	E	EFF
8	1, 1, 1, 0, -1, 1, 0, -1	6, 1, -1, 1, 0, 0, -1, -1	6	0.75
	1, 1, -1, -1, 0, -1, 0, -1	6, 1, 0, 0, 0, 0, -1, -1	6	0.75
	1, 1, -1, 0, 1, -1, 0, -1	6, -1, -1, 1, 0, 0, -1, -1	6	0.75
	1, 0, 1, -1, -1, 0, -1, -1	6, 1, 1, 1, -1, -1, -1, -1	6	0.75
	1, 1, -1, 1, 1, 0, 0, -1	6, 0, -1, 1, 0, 1, -1, -1	6	0.75
	1, -1, 1, 1, 1, 0, 0, -1	6, 0, 1, -1, 0, -1, 1, -1	6	0.75
	1, -1, 1, 0, -1, 0, 1, 1	6, -1, -1, 0, 0, 0, 0, 1	6	0.75
	1, 0, 0, 1, -1, 1, 1, 1	6, 0, 1, 1, 0, 1, 1, 1	6	0.75
	1, 0, -1, 1, 0, 1, 1, 1	6, 1, 1, 1, 0, 0, 1, 1, 1	6	0.75
	1, 1, 0, -1, 1, 1, 0, 1	6, 1, -1, 1, 1, 1, 1, 1	6	0.75
	1, 0, -1, -1, 0, 1, -1, 1	6, -1, -1, -1, 0, 0, -1, 1	6	0.75
	1, 0, 1, 0, 1, -1, -1, 1	6, -1, 0, 0, 0, 0, -1, 1	6	0.75
	1, 0, 1, 1, 1, -1, -1, 1	7, 1, -1, 0, 1, 0, -1, 1	7	0.88
	1, 0, 1, 1, 0, -1, -1, 1	6, 1, -1, -1, 0, 0, -1, 1	6	0.75
	1, 0, 0, 1, -1, -1, -1, 1	6, 0, -1, -1, 0, -1, -1, 1	6	0.75
	1, -1, 0, -1, -1, 1, 0, -1	6, -1, -1, 1, -1, 1, 1, -1	6	0.75
	1, 1, -1, -1, 1, -1, 0, -1	7, -1, -1, 0, 1, 0, -1, -1	7	0.88
	1, 1, 0, -1, 1, -1, 0, -1	6, -1, 1, -1, 1, -1, -1, -1	6	0.75
	1, -1, 0, 1, 1, -1, 0, -1	6, -1, -1, -1, 1, -1, 1, -1	6	0.75
	1, 0, -1, 1, -1, 1, 1, 1	7, -1, 1, 0, -1, 0, 1, 1	7	0.88
	1, -1, -1, 1, 0, 1, 1, 1	7, 1, 0, 1, -1, -1, 0, 1	7	0.88
	1, 0, -1, -1, -1, 0, -1, 1	6, 1, 1, -1, -1, -1, -1, 1	6	0.75

TERNARY BARKER SEQUENCES (Contd)

L	SEQUENCE	AUTOCORRELATION	E	EFF
8	1, 0, -1, -1, 1, 0, 1, 1	6, 1, -1, -1, -1, -1, 1, 1	6	0.75
	1, 0, -1, 1, -1, 0, 1, 1	6, -1, -1, 1, -1, -1, 1, 1	6	0.75
	1, -1, 1, 0, -1, 1, 1, 1	7, -1, 0, 1, -1, 1, 0, 1	7	0.88
	1, 0, 1, 1, -1, 0, -1, 1	6, -1, 1, -1, -1, 1, -1, 1	6	0.75
	1, 0, 1, 1, 1, 0, -1, 1	6, 1, 1, 1, 1, 1, -1, 1	6	0.75
9	1, 1, 1, 1, 1, -1, 0, 0, 1, -1	7, 1, 1, -1, 1, 0, 0, 0, -1	7	0.78
	1, 1, 0, 0, 1, 1, -1, 1, -1	7, -1, 1, 1, 1, 0, 0, 0, -1	7	0.78
	1, -1, 0, 0, 1, -1, -1, -1, -1	7, 1, 1, -1, 1, 0, 0, 0, -1	7	0.78
	1, 0, 0, -1, -1, -1, 1, -1, -1	7, 1, 0, 0, 1, 0, 1, -1, -1	7	0.78
	1, 1, -1, -1, 0, 0, -1, 1, -1	7, -1, -1, 0, 0, -1, 1, 0, -1	7	0.78
	1, 0, 1, 0, 1, -1, -1, 1, 1	7, 0, -1, -1, 1, 0, 0, 1, 1	7	0.78
	1, 0, 1, 0, 1, 1, -1, -1, 1	7, 0, -1, 1, 1, 0, 0, -1, 1	7	0.78
	1, 1, 0, 1, 1, -1, 0, -1, 1	7, 0, 1, 0, 0, 0, -1, 0, 1	7	0.78
	1, -1, -1, -1, 1, -1, 0, 0, -1	7, -1, 0, 0, 1, 0, 1, 1, -1	7	0.78
	1, -1, 0, -1, -1, -1, 1, 0, -1	7, 0, 0, 0, 1, -1, 1, 1, -1	7	0.78
	1, 1, 1, 0, 0, -1, 1, 1, -1	7, 1, -1, 0, 0, 1, 1, 0, -1	7	0.78
	1, 1, 0, 1, -1, 1, 1, 0, -1	7, 0, 0, 0, 1, 1, 1, -1, -1	7	0.78
	1, -1, 1, 0, -1, 0, 1, 1, 1	7, 0, 0, 0, -1, 0, 1, 0, 1	7	0.78
10	1, 1, 0, -1, -1, 0, -1, -1, 1, -1	8, 1, 0, 1, -1, -1, -1, 0, 0, -1	8	0.8
	1, 1, -1, -1, 0, -1, 0, -1, 1, -1	8, -1, 1, -1, 1, -1, -1, 1, 0, -1	8	0.8
	1, 1, 0, 1, -1, -1, -1, 1, 0, -1	8, 1, -1, -1, 0, -1, -1, 1, -1, -1	8	0.8

TERNARY BARKER SEQUENCES (Contd)

L	SEQUENCE	AUTOCORRELATION	E	EFF
10	1,-1,-1,1,1,0,1,1,-1,0, -1	8,-1,-1,-1,-1,-1,1, 0,1,-1	8	0.8
	1,1,1,-1,0,1,0, -1,1,-1	8,-1,-1,1,1,-1,1, -1,0,-1	8	0.8
	1,1,1,-1,-1,0,0, -1,1,-1	8,0,0,-1,-1,-1,1, 1,0,-1	8	0.8
	1,-1,0,1,-1,0,-1, 1,1,1	8,-1,0,-1,-1,1,-1,0,0,1, 1	8	0.8
	1,1,1,0,1,0,1,1,-1,1	8,1,1,1,1,1,-1,1,0,1	8	0.8
	1,0,-1,-1,1,-1,-1,0,-1,1	8,-1,-1,1,0,1,-1,-1, -1,1	8	0.8
	1,0,1,1,0,1,-1,-1,1,1	8,1,-1,1,-1,1,1, 0,1,1	8	0.8
	1,-1,1,1,-1,0,0,1,1,1	8,0,0,1,-1,1,1,1,0,1	8	0.8
	1,-1,1,1,0,-1,0,1,1,1	8,1,-1,-1,1,1,1,1,0,1	8	0.8
	1,1,-1,-1,-1,1,-1,0,0,-1	8,0,-1,-1,0,1,0,1,-1, -1	8	0.8
	1,1,1,0,-1,0,1,-1,1,-1	8,-1,1,-1,-1,1,1,-1, 0,-1	8	0.8
	1,-1,0,-1,1,1,-1,-1,0,-1	8,-1,-1,-1,0,1,1, -1,1,-1	8	0.8
	1,1,1,1,-1,0,-1,1,0,-1	8,1,1,-1,-1,1,-1,0, -1,-1	8	0.8
	1,0,1,-1,0,-1,-1,1,-1,-1	8,-1,1,1,-1,1,-1, 0,-1,-1	8	0.8
	1,1,-1,-1,-1,0,0,-1,1,-1	8,0,0,-1,-1,1,-1,1, 0,-1	8	0.8
	1,-1,0,-1,-1,-1,-1,1,0,-1	8,1,1,1,0,1,-1,1,1,-1	8	0.8
	1,1,0,-1,-1,1,-1,-1,0,-1	8,1,-1,1,0,1,-1,-1, -1,-1	8	0.8

TERNARY BARKER SEQUENCES (Contd)

L	SEQUENCE	AUTOCORRELATION	E	EFF
10	1, 0, 0, -1, -1, -1, 1, -1, -1	8, 0, -1, 1, 0, -1, 0, -1, -1, 1	8	0.8
	1, -1, 1, 0, -1, 0, 1, 1, 1, 1	8, 1, 1, 1, -1, -1, 1, 1, 0, 1	8	0.8
	1, 0, 1, -1, -1, 1, 1, 0, 1, 1	8, 1, -1, 1, 0, -1, 1, 1, 1, 1	8	0.8
	1, 0, -1, -1, 0, -1, -1, 1, -1, 1	8, -1, 1, 1, -1, -1, -1, 0, -1, 1	8	0.8
	1, 0, 1, 1, 0, 1, -1, 1, -1, 1	8, 1, 1, -1, -1, -1, -1, 0, -1, 1	8	0.8
	1, 0, 1, -1, -1, -1, 1, 0, -1, 1	8, -1, -1, -1, 0, -1, -1, 1, -1, 1	8	0.8
	1, -1, -1, 1, -1, 0, 0, 1, 1, 1	8, 0, 0, 1, -1, -1, -1, -1, 0, 1	8	0.8
	1, 0, -1, -1, 1, -1, 1, 0, 1, 1	8, -1, 1, -1, 0, -1, -1, -1, 1, 1	8	0.8
11	1, -1, -1, -1, -1, 1, -1, 0, 0, 0, -1	8, 0, 1, 0, 0, 1, 0, 1, 1, 1, -1	8	0.73
	1, 1, 1, -1, -1, 1, -1, 0, 0, 0, -1	8, 0, -1, 0, 0, -1, 0, 1, -1, -1, -1	8	0.73
	1, 1, 0, -1, 1, 1, 0, 1, -1, 0, -1	8, 0, 0, -1, 0, 1, 0, 1, -1, -1, -1	8	0.73
	1, 1, 0, -1, 0, -1, 0, -1, 1, -1, 0, -1	8, 0, 1, 0, -1, 0, 0, 1, -1, -1, -1	8	0.73
	1, 1, -1, 1, -1, -1, -1, 0, 0, 0, -1	8, 0, 1, 0, 0, -1, 0, -1, 1, -1, -1	8	0.73
	1, -1, 1, 1, -1, -1, -1, -1, 0, 0, -1	8, 0, -1, 0, 0, 1, 0, -1, -1 1, -1	8	0.73
	1, -1, 0, 1, 1, -1, 0, -1, -1, 0, -1	8, 0, 0, 1, 0, -1, 0, -1, -1, 1, -1	8	0.73

TERNARY BARKER SEQUENCES (Contd)

L	SEQUENCE	AUTOCORRELATION	E	EFF
11	1, -1, 0, 1, 0, 1, -1, -1, -1, 0, -1	8, 0, 1, 0, -1, 0, 0, -1, -1, 1, -1	8	0.73
	1, -1, 1, -1, -1, -1, -1, 0, 1, 0, -1	9, 0, 1, -1, -1, 0, 1, 0, 0, 1, -1	9	0.82
	1, -1, -1, 0, 1, -1, 0, -1, -1, 0, -1	8, 0, 0, 1, 1, 1, 1, 0, 0, 1, -1	8	0.73
	1, 1, 1, 1, -1, 1, -1, 0, 1, 0, -1	9, 0, 1, 1, -1, 0, 1, 0, 0, -1, -1	9	0.82
	1, 1, -1, 0, 1, 1, 0, 1, -1, 0, -1	8, 0, 0, -1, 1, -1, 1, 0, 0, -1, -1	8	0.73
	1, 1, 1, 0, -1, 0, 1, 0, -1, 1, -1	8, 0, -1, 0, 0, 0, 1, 0, -1, 0, -1	8	0.73
	1, 1, 1, -1, 0, 0, 1, -1, 1, 0, -1	8, -1, 0, -1, 1, -1, 1, 1, 0, -1, -1	8	0.73
	1, 1, 1, 0, -1, 1, -1, 0, 1, 0, -1	8, 0, -1, 1, -1, -1, 1, 1, 0, -1, -1	8	0.73
	1, 1, -1, -1, 0, -1, -1, 1, -1, 0, -1	9, 0, 0, 1, 0, -1, 1, 1, 0, -1, -1	9	0.82
	1, -1, 1, 1, 0, 0, 1, 1, 1, 0, -1	8, 1, 0, 1, 1, 1, 1, -1, 0, 1, -1	8	0.73
	1, -1, 1, 0, -1, -1, -1, 0, 1, 0, -1	8, 0, -1, -1, -1, 1, 1, -1, 0, 1, -1	8	0.73
	1, -1, -1, 1, 0, 1, -1, -1, -1, 0, -1	9, 0, 0, -1, 0, 1, 1, -1, 0, 1, -1	9	0.82
	1, 0, 0, -1, 0, -1, 1, 1, -1, 1, 1	8, -1, -1, 1, -1, -1, 0, 0, -1, 1, 1	8	0.73
	1, 0, 0, 1, 0, 1, 1, -1, -1, -1, 1	8, 1, -1, -1, -1, 1, 0, 0, -1, -1, 1	8	0.73
	1, 0, -1, 0, 1, 1, 1, 0, 1, -1, 1	8, 0, 1, -1, 1, 1, 1, 1 0, -1, 1	8	0.73

TERNARY BARKER SEQUENCES (Contd)

L	SEQUENCE	AUTOCORRELATION	E	EFF
11	1, 0, -1, 0, 1, -1, 1, 0, 1, 1, 1	8, 0, 1, 1, 1, -1, 1, -1, 0, 1, 1	8	0.73
	1, -1, 0, -1, 1, 1, -1, -1, 0, -1, -1	9, 0, -1, 0, 1, 0, 0, 0, 1, 0, -1	9	0.82
	1, 1, 1, -1, 0, 0, 1, -1, 0 1, -1	8, -1, -1, 0, 1, 0, -1, 1, 0, 0, -1	8	0.73
	1, 1, 1, 0, 1, -1, 0, -1, 1, 1, -1	9, 0, 1, -1, 0, 0, -1, 1, 1, 0, -1	9	0.82
	1, 1, 0, 0, 1, -1, -1, 1, 0, 1, -1	8, -1, -1, 0, 0, 0, -1, 1, 1, 0, -1	8	0.73
12	1, 1, 1, 0, 1, -1, -1, 1, 0, 0, -1	8, 1, 0, 0, 0, 0, -1, 1, -1, -1, -1	8	0.73
	1, 1, 0, -1, -1, 0, 0, -1, -1, 1, -1	8, 1, -1, 0, 1, 0, -1, -1, 0, 0, -1	8	0.73
	1, 1, -1, -1, 0, -1, -1, 0, -1, 1, -1	9, 0, 1, 1, 0, 0, -1, -1, 1, 0, -1	9	0.82
	1, 1, 0, 1, 1, -1, -1, 0, 0, 1, -1	8, 1, -1, 0, 0, 0, -1, -1, 1, 0, -1	8	0.73
	1, -1, 1, 0, 1, 1, -1, -1, 0, 0, -1	8, -1, 0, 0, 0, 0, -1, -1, -1, 0, -1	8	0.73
	1, -1, 0, 0, 1, -1, -1, 1, 1, 1, 1	9, 1, -1, -1, 1, 0, -1, 0, 0, 0, 1	9	0.82
	1, -1, 1, -1, -1, 1, 1, 0, 0, 1, 1	9, -1, -1, 1, 1, 0, -1, 0, 0, 0, 1	9	0.82
	1, 0, 0, -1, 1, 0, -1, 1, 1, 1, 1	8, 1, 0, 1, 0, 0, -1, 0, 1 1, 1	8	0.73
	1, 0, 0, 1, 1, 0, -1, -1, 1, -1, 1	8, -1, 0, -1, 0, 0, -1, 0, 1, -1, 1	8	0.73
	1, 0, -1, 1, 0, 0, -1, 1, 1, 1, 1	8, 1, 0, 0, 1, 0, -1, 1, 0, 1, 1	8	0.73

TERNARY BARKER SEQUENCES (Contd)

L	SEQUENCE	AUTOCORRELATION	E	EFF
12	1, -1, 0, 0, 1, -1, -1, 1, 0 1, 1	8, -1, -1, 0, 0, 0, -1, 1, -1 0, 1	8	0.73
	1, 0, -1, -1, 0, 0, -1, -1 1, -1, 1	8, -1, 0, 0, 1, 0, -1, -1, 0, -1, 1	8	0.73
	1, -1, 0, -1, -1, 1, 1, 0, 0, 1, 1	8, 1, -1, 0, 0, 0, -1, -1, -1, 0, 1	8	0.73
	1, -1, 1, -1, -1, 0, -1, -1, 1, 1, 0, -1	10, -1, -1, 0, -1, -1, 0, 0, 1, 0, 1, -1	10	0.83
	1, -1, 1, 0, 0, -1, 1, 1, -1, -1, -1, -1	10, 0, 0, -1, 1, -1, 0, -1, -1, -1, 0, -1	10	0.83
	1, 1, -1, -1, 0, 1, -1 1, -1, -1, 0, -1	10, -1, -1, 0, -1, 1, 1, 1, -1, 0, -1, -1	10	0.83
	1, 1, 1, -1, 0, -1, 1, -1, -1, 1, 0, -1	10, -1, -1, 0, 1, -1, -1, -1, 1, 0, -1, -1	10	0.83
	1, 0, -1, 1, 1, -1, 0, -1, 1, 1, 1, 1	10, 1, -1, 0, -1, 1, 0, 0, 1, 0, 1, 1	10	0.83
	1, -1, 1, -1, -1, 1, 1, 0, 0, 1, 1, 1	10, 0, 0, 1, 1, 1, 0, -1, -1 1, 0, 1	10	0.83
	1, 0, 1, -1, -1, -1, -1, 0, 1, -1, -1, 1	10, 1, -1, 0, -1, -1, 1, -1, -1, 0, -1, 1	10	0.83
	1, 0, -1, -1, 1, 1, 1, 0, 1, 1, -1, 1	10, 1, -1, 0, 1, 1, -1, 1, 1, 0, -1, 1	10	0.83
13	1, 0, 1, -1, -1, -1, -1, 1, -1, -1, 0, 1, -1	11, 0, 0, 0, 1, 0, -1, 0, -1, 1, -1, 1, -1	11	0.85
	1, 1, 0, -1, 1, 1, 1, -1, 1, -1, -1, 0, -1	11, 0, 0, 0, 1, 0, -1, 0, -1, -1, -1, -1, -1	11	0.85
14	1, -1, 0, -1, 1, 1, 1, 1, -1, 1, 1, -1, 0, 1	12, -1, -1, 1, 0, 1, 0, 1, 0, 1, 1, -1, -1, 1	12	0.86
	1, -1, 0, -1, 1, -1, -1, 1, -1, -1, -1, -1, 0, 1	12, -1, 1, 1, 0, 1, 0, 1, 0, 1, -1, -1, -1, 1	12	0.86

TERNARY BARKER SEQUENCES (Contd)

L	SEQUENCE	AUTOCORRELATION	E	EFF
14	1, -1, 1, -1, 0, 1, 0, -1, -1, 1, 1, 1, 1, 1	12, 1, 1, -1, 0, 0, -1, 1, -1, 0, 0, 1, 0, 1	12	0.86
	1, 1, -1, -1, -1, 1, -1, 1, 1, 0, 1, 1, 0, 1	12, 1, 1, 1, -1, -1, 0, -1, 0, -1, 1, 0, 1, 1	12	0.86
	1, 1, 1, 1, 1, -1, -1, 0, 1, -1 1, -1, 0, 1	12, 1, 1, 0, -1, -1, 1, -1, -1, 0, 1, 0, 1, 1	12	0.86
	1, 0, 1, -1, 0, -1, 1, 1, 1, 1, -1, 1, 1, -1	12, -1, 1, -1, -1, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1, -1	12	0.86
	1, 0, -1, -1, -1, -1, 0, 1, -1, -1, 1, -1, 1, -1	12, -1, 1, 0, -1, 1, 1, 1, -1, 0, 1, 0, 1, -1	12	0.86
	1, 0, -1, -1, 1, 1, 1, -1, 1, -1, -1, 0, 1, -1	12, 1, -1, -1, 0, -1, 0, -1, 0, -1, 1, 1, -1, -1	12	0.86
	1, 0, -1, 1, -1, 1, 1, 1, -1, -1, -1, 0, -1, -1	12, 1, 1, -1, 0, -1, 0, -1, 0, -1, -1, 1, -1, -1	12	0.86
	1, 1, 1, 1, 0, -1, 0, 1, -1, -1, 1, -1, 1, -1	12, -1, 1, 1, 0, 0, -1, -1, 0, 0, -1, 0, -1	12	0.86
15	1, 1, 1, 1, 0, -1, -1, 1, 1, -1, 0, 1, -1, 1, -1	13, 0, -1, 0, -1, 0, 0, 0, 1, 0, 1, 0, -1, 0, -1	13	0.87
	1, 0, 0, -1, -1, -1, -1, 1, -1, 1, -1, -1, 1, 1, -1	13, -1, 0, 0, -1, 0, 1, 0, -1 0, -1, 0, 1, 1, -1	13	0.87
	1, 0, 0, 1, -1, 1, -1, -1, -1, -1, -1, 1, 1, -1, -1	13, 1, 0, 0, -1, 0, 1, 0, -1, 0, -1, 0, 1, -1, -1	13	0.87

APPENDIX B
QUINQUINARY BARKER SEQUENCES

L	SEQUENCE	AUTOCORRELATION	E	EFF
5	1, -1, -2, 0, -1	7, 1, 0, 1, -1	7	0.35
	1, 1, -2, 0, -1	7, -1, 0, -1, -1	7	0.35
6	1, -2, 1, 1, 1, 1	9, -1, 1, 0, -1, 1	9	0.38
	1, -1, 1, -1, -2, -1	9, 1, 1, 0, -1, -1	9	0.38
	1, -2, 1, 2, 1, 1	12, 1, 0, 1, -1, 1	12	0.5
	1, -1, 2, -1, -2, -1	12, -1, 0, -1, -1, -1	12	0.5
7	1, 2, 0, -1, 1, -1, 1	9, -1, 0, 0, -1, 1, 1	9	0.32
	1, -2, 0, 1, 1, 1, 11	9, 1, 0, 0, -1, -1, 1	9	0.32
	1, -1, 1, 2, 0, 0, -1	8, 0, -1, 0, -1, 1, -1	8	0.29
	1, 1, 1, -2, 0, 0, -1	8, 0, -1, 0, -1, -1, -1	"	"
	1, -1, -2, 0, -1, 0, -1	8, 1, 1, 1, 1, 1, -1	"	"
	1, 1, -2, 0, 1, 0, -1	8, -1, 1, -1, 1, -1, -1	"	"
8	1, 2, 1, -1, -1, 2, -1, 1	14, -1, -1, -1, 1, 1, 1, 1	14	0.44
	1, 1, 2, 1, -1, -1, 2, -1	14, 1, -1, 1, 1, -1, 1, -1	14	0.44
	1, 1, 1, 0, -1, -1, 2, -1	10, -1, -1, -1, 0, 0, 1, -1	10	0.31
	1, 2, 1, -1, 0, 1, -1, 1	10, 1, -1, 1, 0, 0, 1, 1	10	0.31
9	1, 0, 2, 0, 2, -1, -2, 1, 1	16, -1, -1, -1, 0, 1, 0, 1, 1	16	0.44
	1, -1, -1, -1, 0, 1, -1, -2, -1	11, 0, 0, 0, 0, 1, 0, -1, -1	11	0.31
	1, 1, 0, 2, 2, -2, 0, -1, 1	16, 0, 0, 0, 0, 0, -1, 0, 1	16	0.44
	1, 1, 1, 1, 0, -1, -1, 2, -1	11, 0, 0, 0, 0, -1, 0, 1, -1	11	0.31
	1, 0, 2, 0, 2, 1, -2, -1, 1	16, 1, -1, 1, 0, -1, 0, -1, 1	16	0.44

QUINQUINARY BARKER SEQUENCES (contd)

L	SEQUENCE	AUTOCORRELATION	E	EFF
10	1, 1, 1, 2, 0, -2, 1, 0, 1, -1	14, 1, 0, -1, 1, 1, 0, 0, 0, -1	14	0.35
	1, -1, 1, -1, -1, -2, -1, 1, 0, -1	12, 1, 1, -1, 1, 1, -1, 0, 1, -1	12	0.3
	1, 1, 1, 1, 1, 0, -1, -1, 2, -1	12, 1, 1, 1, 1, -1-1, 0, 1, -1	12	0.3
	1, 2, 1, -1, 0, 1, -1, 1, -1, 1	12, -1, 1, -1, 1, -1, 0, 1, 1	12	0.3
	1, -1, 1, -2, 0, 2, 1, 0, 1, 1	14, -1, 0, 1, 1, -1, 0, 0, 0, 1	14	0.35
	1, 0, -1, -1, 2, -1, 1, 1, 1, 1	12, -1, 1, 1, 1, -1, -1, 0, 1, 1	12	0.3
11	1, -1, 1, -1, -1, -1, -2, 0, 1, 0, -1	12, 1, 1, 0, -1, 1, 0, 0, 0, 1 -1	12	0.27
	1, -1, 0, 2, -1, 1, 1, 2, 0, 0, -1	14, -1, 1, 1, 1, -1, 0, 0, 0, 1, -1	14	0.32
	1, 1, 1, 1, -1, 1, -2, 0, 1, 0, -1	12, -1, 1, 0, -1, -1, 0, 0, 0, -1, -1	12	0.27
	1, 1, 0, -2, -1, -1, 1, -2, 0, 0, -1	14, 1, 1, -1, 1, 1, 0, 0, 0, -1, -1,	14	0.32
	1, -1, 1, -1, 1, 1, 0, 1, 2, 0, -1	12, -1, 1, 1, 1, -1, 0, 0, 1, 1, -1	12	0.27
	1, 1, 1, 1, 1, -1, 0, -1, 2, 0, -1	12, 1, 1, -1, 1, 1, 0, 0, 1, -1, -1	12	.27
	1, -1, 1, 1, -1, 1, 1, 2, 0, 0, -1	12, 0, 1, 0, 0, 1, 0, 1, -1, 1, -1	12	0.27
	1, 0, 0, 2, -1, 1, 1, 1, -1, -1, -1	12, 0, 1, 0, 0, -1, 0, -1-1, -1, -1	12	0.27
	1, 1, 1, 2, 1, -2, -1, 2, -1, 1, -1	20, 0, -1, 0, 0, 0, 1, 0-1, 0, -1	20	0.45
	1, -1, -1, 0, 2, -1, 0, -1, -2, 0, -1	14, 0, 0, 0, -1, 1, 1, 1, -1, 1, -1	14	0.32
	1, 1, -1, 0, 2, 10, 1, -2 0, -1	14, 0, 0, 0, -1, -1, 1, -1, -1, -1, -1	14	0.32

QUINQUINARY BARKER SEQUENCES (contd)

L	SEQUENCE	AUTOCORRELATION	E	EFF
11	1, 0, 0, -1, 2, -1, -1, 1, 1, 1, 1	12, -1, -1, 1, 1, -1, 0, 0, 1, 1, 1	12	0.27
	1, 0, 0, 1, 2, 1, -1, -1, 1, -1, 1	12, 1, -1, -1, 1, 1, 0, 0, 1, -1, 1	12	0.27
	1, 0, -1, 1, 1, -2, 1, 0, 2, 1, 1	15, -1, 1, 1, 1, -1, 1, 0, 1, 1, 1	15	0.34
	1, 0, -1, -1, 1, 2, 1, 0, 2, -1, 1	15, 1, 1, -1, 1, 1, 1, 0, 1, -1, 1	15	0.34
12	1, -1, 2, -1, 1, 1, 0, 1, 2, 0, -2, -1	19, -1, 1, -1, 0, -1, 0 0, -1, 0, -1, -1	19	0.40
	1, -1, 0, 0, 1, -2, 1, 1, -1, -1, -2, -1	16, 0, 0, 0, 1, -1, 0, 1, 0, 1, -1, -1	16	0.33
	1, 1, 1, 1, 1, 0, -2, 0, 2 -1, 1, -1	16, 0, 0, 0, -1, 1, 0, 1, -1, 0, -1	16	0.33
	1, -2, 0, 2, -1, 0, -1, 1, 1, 2, 1, 1,	19, 1, 1, 1, 0, 1, 0, 0, -1, 0, -1, 1	19	0.4
	1, -2, 1, -1, -1, 1, 2, 1, [illegible], 0, 1, 1	16, 0, 0, 0, 1, 1, 0, -1, 0, -1, -1, 1	16	0.35
	1, -1, 1, -1, 1, 0, -2, 0, 2, 1, 1, 1	16, 0, 0, 0, -1, -1, 0, -1 1, 1, 0, 1	16	0.33
13	1, 1, 1, 1, 1, -1, 2, -2, -1, 1, 1, 0, -1	18, -1, 0, 1, 1, 0, -1, 0, 0, 1, 0, -1, -1	18	0.35
	1, -1, 0, 1, 1, -2, 2, 1, 2, 1, -1, 0, -1	20, -1, 1, -1, 0, 0, -1, 0, 0, 1, -1, 1, -1	20	0.38
	1, -1, 1, -1, 1, 1, 2, 2, -1, -1, 1, 0, -1	18, 1, 0, -1, 1, 0, -1, 0, 0, -1, 0, 1, -1	18	0.35
	1, 1, 0, -1, 1, 2, 2, -1, 2, -1, -1, 0, -1	20, 1, 1, 1, 0, 0, -1, 0, 0, -1, -1, -1, -1	20	0.38
	1, 0, 1, -2, -1, -2, 0, 1, -1, -1, 1, 0, -1	16, 1, -1, -1, 1, -1, 0, 0, 1, 1, 0, 0, -1	16	0.31

QUINQUINARY BARKER SEQUENCES (contd)

L	SEQUENCE	AUTOCORRELATION	E	EFF
13	1, 0, -1, 2, -1, 2, 0, -1, -1 1, 1, 0, -1	16, -1, -1, 1, 1, 1, 0, 0, 1 -1, 0, 0, -1	16	0.31
	1, -1, 2, -1, 0, 1, 0, 0, 1, 1, -1, -2, -1	16, -1, 0, -1, -1, -1, -1, 1, 0, -1, -1, -1, -1	16	0.31
	1, 1, 2, 1, 0, -1, 0, 0, 1, -1, -1, 2, -1	16, 1, 0, 1, -1, 1, -1, -1, 0, 1, -1, 1, -1	16	0.31
	1, -1, 1, -1, 1, 1, -2, -1, 1 1, 1, 1, 1	16, 0, -1, 0, -1, 0, -1, 0, 1, 0, 1, 0, 1	16	0.31
	1, -1, -1, 2, 0, -1, 0, -1, 1, 0, 2, 1, 1	16, 0, 0, 1, -1, -1, -1, 1, 1, -1, 0, 0, 1	16	0.31
	1, 0, -2, -1, 1, 1, 0, 1, 1, 0, 2, -1, 1	16, 0, 0, 0, 1, 1, -1, -1 -1, 1, 0, -1, 1	16	0.31
	1, 0, 2, 1, 2, -1, 0, -1, 0, 2, -1, -1, 1	19, 0, 1, 0, 0, 1, 1, -1, -1, 1, 1, -1, 1	19	0.37

APPENDIX C
QUINQUINARY BROAD BARKER SEQUENCES

L	SEQUENCE	AUTOCORRELATION	E	EFF
5	1, -2, 1, 2, 1	11, 0, -2, 0, -1	5	0.55
6	1, 0, -2, 2, 1, 1	11, -1, -2, 0, 1, 1	5	0.46
	1, -2, 0, 2, 1, 1	11, 1, -2, 0, -1, 1	5	0.46
	1, -1, 2, 2, 0, -1	11, 1, -2, 0, 1, -1	5	0.46
	1, -1, 2, 0, -2, -1	11, -1, -2, 0, -1, -1	5	0.46
7	1, 2, 1, 0, -2, 2, -1	15, -2, 1, -2, 1, 0, -1	7	0.54
	1, -2, 2, 0, -1, -2, -1	15, -2, 1, -2, 1, 0, -1	7	0.54
	1, 2, 2, 0, -1, 2, -1	15, 2, 1, 2, 1, 0, -1	7	0.54
8	1, -1, 2, 0, 2, 2, -1, -1	16, 0, 2, 0, -2, 1, 0, -1	8	0.05
	1, 0, 2, 1, 2, -2, -1, 1	16, 1, 0, -2, 1, 0, -1, 1	8	0.5
	1, 1, -2, -2, 1, -2, 0, -1	16, -1, 0, 2, 1, 0, -1, -1	8	0.5
	1, 0, 2, 1, 2, -1, -1, 1	13, 2, 2, 0, 1, 1, -1, 1	6	0.41
9	1, 1, 2, 2, -2, 0, 0, 1, -1	16, 2, 0, -2, 2, 0, -1, 0, -1	8	0.44
	1, -2, 1, 2, -1, -1, -2, -1, -1	18, 2, -1, 1, -2, 0, -1, 1, -1	9	0.50
	1, 1, 0, 0, 2, 2, -2, 1, -1	16, -2, 0, 2, 2, 0, -1, 0, -1	8	0.44
	1, -2, 2, 0, -1, 0, 2, 2, 1	19, 0, 0, 0, 2, 0, 0, 0, 1	9	0.53
	1, -1, 2, -1, 1, 2, -1, -2, -1	18, -2, -1, -1, -2, 0, -1, -1, -1	9	0.5
	1, 1, 2, 1, 0, -2, 2, 0, -1	16, 1, -1, 1, 2, -1, 0, -1, -1	8	0.44
	1, -1, 2, -1, 1, 2, 0, -2, -1	17, -2, -1, -2, 0, -1, 0, -1, -1	8	0.47
	1, 0, -2, 2, 0, -1, -2, -1, -1	16, 1, -1, 1, 2, -1, 0, -1, -1	8	0.44

QUINQUINARY BROAD BARKER SEQUENCES(contd)

L	SEQUENCE	AUTOCORRELATION	E	EFF
9	1, 1, 2, 1, 1, -2, 0, 2, -1	17, 2, -1, 2, 0, 1, 0, 1, -1	8	0.47
	1, -1, 2, -1, 0, 2, 2, 0, -1	16, -1, -1, -1, 2, 1, 0, 1, -1	8	0.44
	1, -2, 0, 2, -1, -1, -2, -1, -1	17, 2, -1, 2, 0, 1, 0, 1, -1	8	0.47
10	1, 1, -1, -2, -2, 1, -2, 0, 0, -1	17, 2, 1, 1, 0, 1, 0, 1, -1, -1	8	0.43
	1, 2, 2, 0, -2, 1, 1, -2, 2, -1	24, -3, -2, 3, -3, 1, 1, 0, 0, -1	8	0.6
	1, 1, -1, -2, -2, 1, -1, -1, 1, -1	16, 2, -2, 1, -1, 1, -1, 1, 0, -1	8	0.4
	1, -1, -1, 2, 1, -2, 0, -2, 0, -1	17, -2, -2, 1, 1, -1, 0, -1, 1, -1	8	0.43
	1, 0, 2, 0, 2, 1, -2, -1, 1, 1	17, 2, -2, -1, 1, 1, 0, 1, 1, 1	8	0.43
	1, 0, 0, -2, -1, -2, 2, -1, -1, 1	17, -2, 1, -1, 0, -1, 0, -1 -1, 1	8	0.43
	1, -1, -1, 2, -2, -1, -1, 1, 1, 1	16, -2, -2, -1, -1, -1, -1, 0, 1	8	0.4
11	1, 1, 1, -2, -2, -2, 2, -2, 0, 1, -1	25, -1, -1, -2, -2, -2, 0, 1, 0, 0, -1	12	0.57
	1, 1, 0, -2, -2, -2, 2, -2, -1, 1, -1	25, 1, -1, 2, -2, 2, 0, -1, 0, 0, -1	12	0.57
	1, 1, 1, 2, 2, -2, -2, 2, -1, 2, -1	29, -2, 0, -2, -2, 2, 1, 1, 0, 1, -1	14	0.66
	1, -1, 1. -2, 2, 2, -2, -2, -1, -2, -1	29, 2, 0, 2, -2, -2, 1, -1, 0, -1, -1	14	0.66
	1, 1, 1, 2, 2, -2, -2, 2, -1, 1, -1	26, 0, -2, 0, 0, 0, -1, 0, -1, 0, -1	13	0.59
12	1, 2, 2, 1, -1, -1, 0, 1, 0, -2, 2, -1	22, 2, 0, -2, -1, 1, -1, 0, -1, 0, 0, -1	11	0.46

QUINQUINARY BROAD BARKER SEQUENCES (contd)

L	SEQUENCE	AUTOCORRELATION	E	EFF
12	1, -1, 2, -1, 1, 1, 0, 2, 2, 0, -2, -1	22, 1, 2, -1, -2, 1, -1 1, -1, 0, -1, -1	11	0.46
	1, -2, 0, 2, -2, C, -1, 1, 1, 2, 1, 1	22, -1, 2, 2, -2, -1, -1, -1, -1, -1, 0, -1, 1	11	0.46
13	1, -1, 1, -1, 2, -1, -2, 2, 2, 1, 1, 0, -1	24, -2, -2, -1, -2, 0, 1, 1, 0, 1, 0, 1, -1	12	0.46
	1, -1, 1, 1, -2, 1, -2, -2, -2, -2, 1, 1, 0, -1	24, 0, 0, -1, -2, 0, -1, 1, 0, -1, 0, 1, -1	12	0.46
	1, 1, -1, -2, 0, 2, 2, 0, 1, 2, -2, 2, -1	29, -2, -1, -2, -1, 2, -1, 1, 1, 0, 1, 1, -1	14	0.56
	1, 1, 1, -1, -2, -1, -2, 2, -2, -1, 1, 0, -1	24, 0, 0, 1, -2, 0, -1, -1, 0, 1, 0, -1, -1	12	0.46
	1, 1, 1, 1, 2, 1, -2, -2, 2 -1, 1, 0, -1	24, 2, -2, 1, -2, 0, 1, -1 0, -1, 0, -1, -1	12	0.46
	1, -1, -1, 2, 0, -2, 2, 0, 1, -2, -2, -2, -1	29, 2, -1, 2, -1, -2, -1, -1, 1, 0, 1, -1, -1	12	0.56
	1, -2, 2, -1, 1, 1, -2, -1, 2 2, 1, 1, 1	28, -2, 1, 1, -1, 1, 2, 0, 0, 1, 1, -1, 1	14	0.56
	1, 2, 2, 1, 1, -1, -2, 1, 2 -2, 1, -1, 1	28, 2, 1, -1, -1, -1, 2, 0, 0, -1, 1, 1, 1	14	0.54
	1, 0, -2, -2, 1, 2, 0, 1, 2, -1, 2, -1, 1	26, -1, 0, -2, 0, 0, -2, 0, 1 -1, 0, -1, 1	13	0.5
	1, 0, -2, 2, 1, -2, 0, -1 2, 1, 2, 1, 1	26, 1, 0, 2, 0, 0, -2, 0, 1, 1, 0, 1, 1	13	0.5
	1, 0, 1, 1, 2, -1, 2, 2, -2, -1, -1, 1, 1	24, 2, -2, 1, 1, 2, -2, 1, 0, 1, 0, 1, 1	12	0.46
	1, 0, 1, -1, 2, 1, 2, -2, -2, 1, -1, -1, 1	24, -2, -2, -1, 1, -2, -2, -1, 0, -1, 0, -1, 1	12	0.46

APPENDIX D

BROAD HUFFMAN SEQUENCES (Quinquinary)

L	SEQUENCE	AUTOCORRELATION	E	EFF
15	1, 1, 1, 0, 1, 1, 1, -2, -1, -1, 2 -1, 0, 1, -1	19, 0, 0, -2, 2, 0, -2, -2, 0, 0, 0, 0, 0, 0, -1	9	0.32
	1, 1, 1, 0, 1, 1, 1, -1, -1, -1, 2, -1, 0, 1, -1	16, 0, 0, 1, 1, 1, 0, -2, 0, 0, 0, 0, 0, 0, -1	8	0.27
	1, 1, 0, -1, -2, -1, 1, -2, -1 1, -1, 0, -1, 1, -1	19, 0, 0, 2, 2, 0, -2, 2, 0, 0, 0, 0, 0, 0, -1	9	0.32
17	1, -2, 2, -2, 2, 0, -2, 1, 2, -1, -2, 0, 2, 2, 2, 2, 1	48, 0, 3, 0, 4, 0, -4, 0, 4, 0, 0, 0, 0, 0, 0, 0, 1	12	0.71
19	1, 1, 1, 0, -2, -2, -2, -1, 1, -2, 1, -2, -1, 2, -1, -1, 0, 1, -1	35, 3, 3, 4, -5, 2, 0, 3, -2, 0, 0, 0, 0, 0, 0, 0, 0, 0, -1	7	0.46
32	1, 0, -2, 1, 2, -2, -1, 1, 1, 2, -2, -2, 2, -2, 2, 2, 2, 2, 1, -1, -2, -2, -1, 1, -1, 2, -2, 2, -1, 2, 0, 1	84, -7, 8, -12, -8, -12, -9, -14, 14, 6, -8, 10, -13, 8, -1, 13, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1	6	0.65

APPENDIX E

INTEGER SEQUENCES Co = C_N ELEMENTS 0, ±1, ±--- ± 7

L	SEQUENCE	AUTOCORRELATION	E	EFF
5	1, -3, 4, 3, 1	36, 0, -1, 0, 1	36	0.45
	1, -3, 5, 3, 1	45, 0, 1, 0, 1	45	0.36
	1, -1, -7, 1, -1	53, -2, -1, 2-1	26	0.22
	1, 1, -7, -1, -1	53, 2, -1, -2, -1	26	0.22
	1, -3, 5, 3, 1	56, 0, 3, 0, 1	18	0.31
6	1, -5, 6, 4, 2, 1	83, -1, 2, 0, -3, 1	27	0.38
	1, -5, 5, 4, 2, 1	72, 0, -1, -1, -3, 1	24	0.48
	1, 2, 3, 3, -4, 1	40, 1, 0, -2, -2, 1	20	0.42
	1, -2, 4, -6, -5, -1	83, 1, 2, 0, -3, -1	27	0.38
	1, -2, 4, -5, -5, -1	72, 0, -1, 1, -3, -1	24	0.48
	1, 4, 3, -3, 2, -1	40, -1, 0, 2, -2, -1	20	0.42
7	1, 3, 3, 6, -5, 0, 1	81, 0, 1, -3, -2, 3, 1	27	0.32
	1, -3, 3, -6, -5, 0, 1	81, 0, 1, 3, -2, -3, 1	27	0.32
	1, 3, 4, 1, -4, 3, -1	53, 0, -2, 0, 1, 0, -1	26	0.47
	1, 4, 7, 5, -7, 4, -1	157, 0, 5, 0, 2, 0, -1	31	0.46
	1, -4, 7, -4, -7, -4, -1	148, 0, -3, 0, 2, 0, -1	49	0.43
	1, 3, 5, 3, -5, 3, -1	79, 0, 3, 0, -1, 0, -1	26	0.45
	1, 3, 6, 4, -6, 3, -1	108, 0, 0, 0, -3, 0, -1	36	0.43
8	1, 4, 6, 1, -5, 4, -2, 1	100, -1, -2, -2, 0, 2, 2, 1	50	0.35
	1, 1, -2, -7, -6, 5, -4, 1	133, 1, -3, -1, 0, -1, -3, -1	44	0.34
	1, -3, 5, -5, 1, 7, 5, 1	136, -1, 2, 3, 0, -3, 2, 1	45	0.35

INTEGER SEQUENCES $|C_0| = |C_N|$ ELEMENTS $\{0, \pm 1, \pm, ---- \pm 7\}$

L	SEQUENCE	AUTOCORRELATION	E	EFF
8	1, 1, -2, -5, -5, 4, -4, 1	100, 1, 2, 0, 1, -2, -3, 1	33	0.35
	1, 1, 3, 3, 7, -5, -1, 1	96, 3, 0, -1, 2, -3, 0, 1	32	0.24
	1, -4, 5, -7, -7, -2, 1, 1	146, 3, 3, -3, -1, -1, -3, 1	48	0.37
	1, 4, 7, 2, -6, 4, -2, 1	127, 0, -3, -4, -2, 3, 2, 1	31	0.32
	1, 2, 3, 3, 6, -6, 0, 1	96, -1, 3, 3, -3, -3, 2, 1	32	0.33
	1, 4, 5, 6, -7, 2, 1, -1	133, -1, -3, 1, 0, 1, -3, -1	44	0.34
	1, 3, 5, 5, 1, -7, 5, -1	136, 1, 2, -3, 0, 3, 2, -1	45	0.35
	1, -4, 5, -1, -5, -4, -2, -1	100, 1, -2, 2, 0, -2, 2, -1	50	0.35
	1, 4, 4, 5, -6, 2, 1, -1	100, -1, 2, 0, 1, 2, -3, -1	33	0.35

APPENDIX F

INTEGER SEQUENCES $|C_0| \neq |C_n|$ ELEMENTS $\{0, \pm 1 ---- \pm 7\}$

L	SEQUENCE	AUTOCORRELATION	E	EFF
4	1, 3, 6, -7	68, 3, -2, -3	22	0.35
	1, 3, 6, -3	55, 3, -3, -3	18	0.38
	1, -2, 5, 2	34, -2, 1, 2	17	0.34
	1, -2, 6, 2	45, -2, 2, 2	22	0.31
	1, -2, 7, 2	58, -2, 3, 2	19	0.30
	1, 2, 5, -2	34, 2, 1, -2	17	0.34
	1, 2, 6, -2	45, 2, 2, -2	22	0.31
	1, 2, 7, -2	58, 2, 3, -2	19	0.30
	1, -3, 7, 3	68, -3, -2, 3	22	0.35
	1, -3, 6, 3	55, -3, -3, 3	18	0.38
5	1, 1, 3, 7, -4	76, -3, -2, 3, -4	19	0.31
	1, -1, 3, -7, -4	76, 3, -2, -3, -4	19	0.31
	2, 5, 6, -5, 2	94, 0, -1, 0, 4	23	0.52
	2, 5, 7, -5, 2	107, 0, 3, 0, 4	26	0.44
	1, 1, 2, 6, -3	51, -3, 2, 3, -3	17	0.28
	1, -1, 2, -6, -3	56, -4, 0, -3, -3	17	0.28
6	1, -3, 6, -7, -7, -2	148, 0, -1, 2, -1, -2	74	0.50
	1, -3, 5, -5, -6, -2	100, -1, 0, 3, 0, -2	33	0.46
	1, -3, 6, -6, -7, -2	135, -1, -6, 3, -1, -2	22	0.46
	1, 2, -1, -7, 1, -2	60, -2, -2, -3, -3, -2	20	0.20
	1, 3, 5, 4, -6, 3	96, -4, -1, 1, 3, 3	24	0.44

INTEGER SEQUENCES $|C_o| \neq |C_n|$ (Contd)

L	SEQUENCE	AUTOCORRELATION	E	EFF
6	1, 3, 6, 5, -7, 3	129, -5, -6, 2, 2, 3	21	0.44
	1, 3, 5, 5, -6, 3	105, -5, 5, 2, 3, 3	21	0.49
	1, 3, 6, 6, -7, 3	140, -6, 0, 3, 2, 3	23	0.48
	2, 4, 6, 5, -7, 3	139, 6, 5, 0, -2, 6	23	0.47
	2, -4, 6, -5, -7, -3	139, -6, 5, 0, -2, -6	23	0.47
	1, -3, 5, -4, -6, -3	96, 4, -1, -1, 3, -3	24	0.44
	1, -3, 6, -5, -7, -3	129, 5, -6, -2, 2, -3	21	0.44
	1, -3, 5, -5, -6, -3	105, 5, 5, -2, 3, -3	21	0.49
	1, -3, 6, -6, -7, -3	140, 6, 0, -3, 2, -3	23	0.48
	1, -2, -1, 7, 1, 2	60, 2, -2, 3, -3, 2	20	0.2
	1, -1, -3, 7, 2, 2	68, -1, -2, -1, 0, 2	34	0.23
	1, 0, -3, 5, 2, 2	43, -1, 1, -1, 2, 2	21	0.29
	1, 0, -4, 7, 3, 2	79, -1, -2, -1, 3, 2	26	0.27
	1, 3, 5, 6, -6, 2	111, 0, 5, -2, 0, 2	22	0.51
	1, 3, 6, 7, -7, 2	148, 0, -1, -2, -1, 2	74	0.50
	1, 3, 5, 5, -6, 2	100, 1, 0, -3, 0, -2	33	0.46
	1, 3, 6, 6, -7, 2	135, 1, -6, -3, -1, 2	22	0.46
	1, 1, -3, -7, 2, -2	68, 1, -2, 1, 0, -2	34	0.23
	1, 0, -3, -5, 2, -2	43, 1, 1, 1, 2, -2	21	0.29
	1, 0, -4, -7, 3, -2	79, 1, -2, 1, 3, -2	26	0.27
	1, -3, 5, -6, -6, -2	111, 0, 5, 2, 0, -2	22	0.51

INTEGER SEQUENCES $|C_0| \neq |C_n|$ (Contd)

L	SEQUENCE	AUTOCORRELATION	E	EFF
7	1, 1, -1, -5, -7, 4, -3	102, 0, 2, -1, 0, 1, -3	34	0.30
	1, -1, -1, 5, -7, -4, -3	102, 0, 2, 1, 0, -1, -3	34	0.30
	1, 1, -1, -5, -6, 4, -3	89, -1, -2, 0, 1, 1, -3	29	0.35
	1, -1, -1, 5, -6, -4, -3	89, 1, -2, 0, 1, -1, -3	29	0.35
	1, 1, 0, -5, -7, 4, -3	101, -4, -4, 3, -3, 1, -3	25	0.29
	1, -1, 0, 5, -7, -4, -3	101, 4, -4, -3, -3, -1, -3	25	0.29
	1, -2, 3, -4, 3, 6, 2	79, -2, 2, 0, -3, 2, 2	26	0.31
	1, -2, 3, -5, 5, 7, 2	117, 1, 3, -4, -3, 3, 2	29	0.34
	1, 2, 3, 4, 3, -6, 2	79, 2, 2, 0, -3, -2, 2	26	0.31
	1, 2, 3, 5, 5, -7, 2	117, -1, 3, 4, -3, -3, 2	29	0.34
	1, -2, 4, -5, 3, 7, 3	113, -3, 0, 2, 1, 1, 3	37	0.33
	1, 2, 4, 5, 3, -7, 3	113, 3, 0, -2, 1, -1, 3	37	0.33
	1, 2, 3, 4, 2, -6, 3	79, -2, -1, 2, -1, 0, 3	26	0.31
	1, -2, 3, -4, 2, 6, 3	79, 2, -1, -2, -1, 0, 3	26	0.31
8	1, -1, -1, -2, 7, 1, 1, 2	62, -2, 1, 2, 1, -2, -1, 2	31	0.16
	1, -3, 6, -6, 0, 7, 6, 2	171, -3, -4, 0, 3, 1, 0, 2	42	0.44
	1, 1, 4, 2, 7, -5, -2, 2	104, -2, 0, -1, -2, 1, 0, 2	52	0.27
	1, -1, -1, -3, 7, 2, 1, 2	70, 0, 0, -1, -2, -1, -1, 2	35	0.18
	1, -3, 5, -5, 0, 6, 5, 2	125, -3, 2, 0, -3, 1, -1, 2	41	0.43
	1, 1, -1, 2, 7, -1, 1, -2	62, 2, 1, -2, 1, 2, -1, -2	31	0.16
	1, 3, 6, 6, 0, -7, 6, -2	171, 3, -4, 0, 3, -1, 0, -2	42	0.44

INTEGER SEQUENCES $|C_0| \neq |C_n|$ (Contd)

L	SEQUENCE	AUTOCORRELATION	E	EFF
8	1, 1, -1, 3, 7, -2, 1, -2	70, 0, 0, 1, -2, 1, -1, -2	35	0.18
	1, -1, 4, -2, 7, 5, -2, -2	104, 2, 0, 1, -2, -1, 0, -2	52	0.27
	1, -1, 0, 3, -7, 6, 6, 4	148, -4, -3, 0, -1, 0, 2, 4	37	0.38
	1, 1, 0, -3, -7, -6, 6, -4	148, 4, -3, 0, -1, 0, 2, -4	37	0.38
	1, 1, 0, -3, -7, -7, 6, -4	161, 5, 4, 0, -2, -1, 2, -4	32	0.41

```
      C       THIS PROGRAM GENERATES INTEGER HUFMAN SEQUENCES
      C       C1 = 1, C(N) = -1

              DOUBLEPRECISION AJ,A,C,B,EFFI,RATIO
              DIMENSION AJ(192),C(192),A(192),B(192),IC(192),
              1IB(192)
              COMMON C,A,B
              N=10

      C       CALCULATING COEFFICIENTS

      5       DO 40 M=1,4
              C(1)=1
              C(2)= 2*M
              C(3)=2*(M**2)
              DO 10 K=4,(N/2)
              C(K)=M*C(K-1)+C(K-2)
      10      CONTINUE

      C       CALCULATING MIDDLE COEFFICIENT

              AJ(1)=M
              AJ(3)=M**3+3*M
              DO 11 IX=5,N/2,2
      11      AJ(IX)=(AJ(1)**2+2)*(AJ((IX)-2))-AJ((IX)-4)
              C(N/2+1)=M*C(N/2)+C((N/2)-1)-C(1)*(AJ(N/2))
      12      DO 20 K=1,(N/2)
              C(N+2-K)=(-C(K))*(-1)**(K-1)
      20      CONTINUE

      C       CALCULATING SECOND HALF COEFFCIENTS

              DO 22 NN=2,N,2
              C(NN)=(-1)*C(NN)
      22      CONTINUE

      C       CALCULATING ENERGY

              E=0
              DO 30 L=1,(N+1)
              E=E+C(L)**2
      30      CONTINUE
              ID=N+1

      C       FINDING MAX ELEMENT,ENERGY RATIO,EFFICIANCY

              CALL MAXC(ID,Z)
              RATIO=E/ABS(C(1)*C(N+1))
              ERATIO=E/(Z)**2
              EFFI=E/((N+1)*(Z**2))
              MD=N+1
              DO 35 L=1,MD
              A(L)=C(L)
              B(L)=0
      35      CONTINUE

      C       FINDING AUTOCORELATION

              CALL AUTO(MD)
              DO 37 I=1,N+1
```

```
      IC(I)=C(I)
      IB(I)=B(I)
37    CONTINUE
      PRINT100,N,M,RATIO,ERATIO,EFFI
100   FORMAT(1H0,' N = ',I5,'M =',I5,'RATIO=',D30.24,
     1'ERATIO=',F10.3,'EFFI=',F6.3)
      PRINT 300, (IC(I),IB(I),I=1,N+1)
300   FORMAT (1H0,'C=',5X,I10,5X,'B=',I10)
40    CONTINUE
      N=N+4
      IF(N.EQ.38)GO TO 200
      GO TO 5
200   STOP
      END
      SUBROUTINE MAXC(KO,BIG)
      COMMON C
      DIMENSION C(192)
      BIG=C(1)
      DO 10 I=1,KO
      IF(BIG.GT.C(I+1))GO TO 10
      BIG=C(I+1)
10    CONTINUE
      RETURN
      STOP
      END
      SUBROUTINE AUTO(NO)
      DOUBLEPRECISION C,A,B
      COMMON C,A,B
      DIMENSION C(192),A(192),B(192)
      DO 20 I=1,NO
      K=NO+1-I
      L=0
      DO 10 J=1,I
      L1=K+L
      B(J)=A(K)*C(L1)+B(J)
      L=L+1
10    CONTINUE
20    CONTINUE
      RETURN
      STOP
      END
```

```
C       THIS PROGRAM GENERATES INTEGER HUFFMAN SEQUENCES
C        C1 = 1, C(N) = 1

         COMPLEX C,A,B,X,
         DIMENSION AJ(192),D(192),C(192),A(192),B(192)
         COMMON D,C,A,B
         N=14

C        CALCULATING COEFFICIENTS

5        DO 40 M=1,4
         C(1)=(1,0)
         C(2)= (0,-2)*M
         C(3)=(-2 ,0)*M**2
         DO 10 K=4,(N/2)
         C(K)=(-C(K-2))-(0,1)*M*C(K-1)
10       CONTINUE

C        CALCULATING MIDDLE COEFFICIENT

         AJ(1)=M
         AJ(3)=M**3+3*M
         DO 11 IX=5,N/2,2
11       AJ(IX)=(AJ(1)**2+2)*(AJ((IX)-2))-AJ((IX)-4)
         C(N/2+1)=-C((N/2)-1)  -(0,1)*M*C(N/2)
         1-C(1)*AJ(N/2)*(0,1)

C        CALCULATING SECOND HALF COEFFICIENTS

12       DO 20 K=1,(N/2)
         C((N/2)+1+K)=C((N/2)+1-K)
20       CONTINUE
         DO 25 I2=1,N+1
         D(I2)=CABS(C(I2))
25       CONTINUE

C        CALCULATING ENERGY

         E=0
         DO 30 L=1,(N+1)
         E=E+D(L)**2
30       CONTINUE
         ID=N+1

C        FINDING MAX ELEMENT,ENERGY RATIO,EFFICIANCY

         CALL MAXC(ID,Z)
         PRINT*,Z
         RATIO=E/(C(1)*C(N+1))
         ERATIO=E/(Z)**2
         EFFI=E/((N+1)*(Z**2))
         MD=N+1
         DO 35 L=1,MD
         A(L)=C(L)
         B(L)=(0,0)
35       CONTINUE
         DO 34 KK=1,N+1
         X=C(KK)
         C(KK)=CONJG (X)
34       CONTINUE
```

```
C        FINDING AUTOCORELATION

         CALL AUTO(MO)
         PRINT100,N,M,RATIO,ERATIO,EFFI
100      FORMAT(1H0,' N = ',I5,'M =',I5,'RATIO=',D30.24,
        1'ERATIO=',F10.3,'EFFI=',F6.3)
         PRINT 300, ( C(I), B(I),I=1,N+1)
300      FORMAT (1H0,'C=',2G20.8,5X,'B=', 2G20.8)
40       CONTINUE
         N=N+8
         IF(N.EQ.88)GO TO 200
         GO TO 5
200      STOP
         END
         SUBROUTINE MAXC(KO,BIG)
         COMMON D
         DIMENSION D (192)
         BIG=D(1)
         DO 10 I=1,KO
         IF(BIG.GT.D(I+1))GO TO 10
         BIG=D(I+1)
10       CONTINUE
         RETURN
         STOP
         END
         SUBROUTINE AUTO(NO)
         COMPLEX C,A,B
         COMMON D,C,A,B
         DIMENSION D(192), C(192),A(192),B(192)
         DO 20 I=1,NO
         K=NO+1-I
         L=0
         DO 10 J=1,I
         L1=K+L
         B(J)=A(K)*C(L1)+B(J)
         L=L+1
10       CONTINUE
20       CONTINUE
         RETURN
         STOP
         END
```

```
C          THIS PROGRAM GENERATES INTEGER HUFMAN SEQUENCES
C           C1 > 1, C(N) < -1

            DOUBLEPRECISION AJ,A,C,B,EFFI,RATIO,Z
            DIMENSION AJ(192),C(192),A(192),B(192)
            COMMON C,A,B
            N=10

C           CALCULATING COEFFICIENTS

5           DO 40 M=1,4
            W=M
            X=4./W
            C(1)=1.
            C(2)=2./X
            C(3)=2./(X**2)
            DO 10 K=4,(N/2)
            C(K)=(1./X)*C(K-1)+C(K-2)
10          CONTINUE

C           CALCULATING MIDDLE COEFFICIENT

            AJ(1)=(1./X)
            AJ(3)=(1./X)**3+3*(1./X)
            DO 11 IX=5,N/2,2
11          AJ(IX)=(AJ(1)**2+2)*(AJ((IX)-2))-AJ((IX)-4)

C           CALCULATING SECOND HALF COEFFCIENTS

            C(N/2+1)=(1./X)*C(N/2)+C((N/2)-1)-C(1)*(AJ(N/2))
12          DO 20 K=1,(N/2)
            C(N+2-K)=(-C(K))*(-1)**(K-1)
20          CONTINUE
            DO 25 JJ=1,N+1
            C(JJ)=C(JJ)*4**(N/2)
25           CONTINUE

C           CALCULATING ENERGY

            E=0
            DO 30 L=1,(N+1)
            E=E+C(L)**2
30          CONTINUE

C           FINDING MAX ELEMENT,ENERGY RATIO,EFFICIANCY

            ID=N+1
            CALL MAXC(ID,Z)
             PRINT *,Z
            RATIO=E/ABS(C(1)*C(N+1))
            ERATIO=E/(Z)**2
            EFFI=E/((N+1)*(Z**2))
            MO=N+1
            DO 35 L=1,MO
            A(L)=C(L)
            B(L)=0
35          CONTINUE

C           FINDING AUTOCORELATION

            CALL AUTO(MO)
```

```
C     THIS PROGRAM GENERATES HUFFMMAN SEQUENCES
      1L=2**(N)-1

      DOUBLEPRECISION AJ,A,C,B,EFFI,RATIO,Z
      DIMENSION AJ(192),C(192),A(192),B(192)
      COMMON C,A,B
C     CALCULATING COEFFICIENTS

      X=.5
5     DO 40 M=3,5
      N=2**M-1
      C(1)=1
      C(2)=X-X**(-1)
      DO 10 K=3,N
      C(K)=X**(K-1)-X**(K-3)
10    CONTINUE
      C(N+1)=-(X**(N-2))

C     CALCULATING ENERGY

      E=0
      DO 30 L=1,(N+1)
      E=E+C(L)**2
30    CONTINUE
      IO=N+1
      CALL MAXC(IO,Z)
      PRINT*,Z
      RATIO=E/ABS(C(1)*C(N+1))
      ERATIO=E/(Z)**2
      EFFI=E/((N+1)*(Z**2))
      MO=N+1
      DO 35 L=1,MO
      A(L)=C(L)
      B(L)=0
35    CONTINUE

C     CALCULATING AUTOCORRELATION

      CALL AUTO(MO)
      PRINT100,N,M,RATIO,ERATIO,EFFI
100   FORMAT(1H0,' N=',I5,',  M=',I5,',  RATIO=',D30.
     124,',  ERATIO=',F10.3,',  EFFI=',F6.3)
      PRINT36,(C(I),B(I),I=1,N+1)
36    FORMAT(1H0,'C=',G20.8,'   B=',G20.8)
40    CONTINUE
      X=4.*X
      IF(X.GT.(3.0)) GO TO 200
      GO TO 5
200   STOP
      END
      SUBROUTINE MAXC(KO,BIG)
      DOUBLEPRECISION C,BIG
      COMMON C
      DIMENSION C(192)
      BIG=C(1)
      DO 10 I=1,KO
      IF(DABS(BIG).GT.DABS(C(I+1)))GO TO 10
      BIG=C(I+1)
10    CONTINUE
```

```
      RETURN
      STOP
      END
      SUBROUTINE AUTO(ND)
      DOUBLEPRECISION C,A,B
      COMMON C,A,B
      DIMENSION C(192),A(192),B(192)
      DO 20 I=1,ND
      K=ND+1-I
      L=0
      DO 10 J=1,I
      L1=K+L
      B(J)=A(K)*C(L1)+B(J)
      L=L+1
   10 CONTINUE
   20 CONTINUE
      RETURN
      STOP
      END
```

```
C      THIS PROG GENERATES INTEGER SEQUENCES
C       WITH MAG(C1,CN)  > 1

        COMMON ISOL(64,6),LIMIT,IRHS,ICOEFF(6),INOVAR,
       1ID(64),INY(99)
        DIMENSION INSOL(6,2),IOUTY(99),IYSOL(8000,3),
       1IA(32,729)

C       ISOL(64,6) CONTAINS SOLUTIONS RETURNED BY SUB.
C       SOLVE

C       'LIMIT' IS THE NUMBER OF ABOVE SOLUTIONS

C       'IRHS' IS THE RIGHT HAND SIDE OF BOOLEAN EQN.

C       ICOEFF(6) CONTAINS COEFFICIENTS BEING
C       PASSED TO SOLVE

C       INOVAR IS THE NUMBER OF VARIABLES BEING PROCESSED
C       IN SOLVE

C       ID(64) CONTAINS A READY SOLUTION FOR FINDING
C       AUTOCORELATION

C       INY(99) CONTAINS THE INPUT VALUES FOR CURRENT
C       SOLUTION

C       INSOL(6,2) CONTAINS THE INITIALLY ASSUMED VALUES
C       OF END COEFFICIENTS

C       IOUTY(99) CONTAINS THE OUTPUT VALUES OF CURRENT
C       SOLUTIONS

C       IYSOL(8000,3) CONTAINS PACKED SOLUTIONS OF ONE
C       ITERATIONS

C       'IA' IS THE SPECIFIED GIVEN AUTOCORRELATION

        READ*,N,K,IP,ITR,LRATIO,IZLIM

C       N IS THE LENGTH OF SEQUENCE

C       'K' IS THE NUMBER OF BINARY BITS REQD TO REPRESEN
C       2*IP

C       'IP'IS THE VALUE OF MAX PERMISSIBLE ELEMENT VALUE

C       'ITR' IS THE NUMBER OF ITERATIONS

C       'LRATIO' IS RATIO OF MAX.LOBE TO SIDE LOBES

C       'IZLIM' IS THE MAX. NO. OF ZERO ELEMENTS ALLOWED
C       IN A SEQUENCE

        READ*,((INSOL(I,J),I=1,6),J=1,2)
        READ*,((IA(I,J), I=1,N-1),J=1,ITR)
        DO 280 INS=2,2

C       TRANSFER INITIAL VALUES OF END COEFFCIENTS INTO
C       'INY'
```

```
      DO 10 J=1,6
      K2=J
      IF(J.GT.3) K2=K2+3*N-6
      INY(K2)=INSOL(J,INS)
10    CONTINUE

C     CALCULATE COEFFICIENTS TO BE PASSED TO SUBROUTI
C     SOLVE

      DO 40 I1=1,K
      I=I1-1
      ICOEFF(K+I1)=0
      DO 20 I2=1,K
      I21=I2-1
      ICOEFF(K+I1)=ICOEFF(K+I1)+(2**I21)*INY(I2)
20    CONTINUE
      ICOEFF(K+I1)=(ICOEFF(K+I1)-IP)*(2**I)
      ICOEFF(I1)=0
      DO 30 I2=1,K
      I21=I2-1
      IQ2=K*(N-1)+I2
      ICOEFF(I1)=ICOEFF(I1)+(2**I21)*INY(IQ2)
30    CONTINUE
      ICOEFF(I1)=(ICOEFF(I1)-IP)*(2**I)
40    CONTINUE
      DO 270 ITI=1,ITR
      ICOUNT=0
      IT=N-1
      INOVAR=6
      ICEND=1

C     ICEND IS A POINTER INDICATING AT ANY TIME THE
C     OF ENTERIES IN IYSOL

      ICPTR=1

C     'ICPTR' IS THE POINTER TO CURRENT SOL.BEING
C     PROCESSED

      ITLIM=(N+1)/2
50    IT=IT-1
      IPEND=ICEND

C     'IPEND' IS THE POINTER TO THE END OF ENTRIES IN
C     IYSOL DUE TO CURRENT SOL

C     CALCULATE 'IRHS'

60    IRHS=IA(IT,ITI)-(N-IT)*IP*IP
      IRHS2=0
      DO 100 I1=1,K
      I=I1-1
      IRHS1=0
      IQ1=I+1
      IRHS1=IRHS1+IP*INY(IQ1)
      IF((N-IT).EQ.2)GO TO 90
      ISUM1=0
      DO 80 J=2,N-IT-1
      ISUM=0
      DO 70 IL=1,K
      IL1=IL-1
```

```
            IQ1=(J-1)*K+1+IL1
            IQ2=IQ1+K*IT
            ISUM=ISUM+(2**IL1)*INY(IQ2)
70          CONTINUE
            IQ1=(J-1)*K+1+I
            ISUM=ISUM*INY(IQ1)
            IQ2=IQ1+K*IT
            ISUM=ISUM-IP*(INY(IQ1)+INY(IQ2))
            ISUM1=ISUM1+ISUM

80          CONTINUE
            IRHS1=IRHS1-ISUM1
90          J=N-IT

            IQ1=(J-1)*K+1+I
            IQ2=IQ1+K*IT
            IRHS1=IRHS1+IP*INY(IQ2)
            IRHS1=(2**I)*IRHS1
            IRHS2=IRHS2+IRHS1
100         CONTINUE
            IRHS=IRHS+IRHS2
999         CALL SOLVE
1000        ICOUNT=ICOUNT+1
115         IF (LIMIT.EQ.0) GO TO 250
            IZCNT=0
            I11=0
            IX1=0
            IQ1=(N-IT-1)*K

C           COPY PREVIOUS LEADING ELEMENTS

            DO 120 I1=1,IQ1
            IOUTY(I1)=INY(I1)
            I11=I11+1
            IF(I11.GT.K) I11=MOD(I11,K)
            IX1=INY(I1)*(2**(I11-1))+IX1
            IF(I11.NE.K) GO TO 120
            IF(IX1.EQ.IP) IZCNT=IZCNT+1
            IX1=0
120         CONTINUE
            I11=0
            IX1=0
            IQ2=K*IT+4

C           COPY PREVIOUS FOLLOWING ELEMENTS

            DO 130 I1=IQ2,3*N
            IOUTY(I1)=INY(I1)
            I11=I11+1
            IF(I11.GT.K) I11=MOD(I11,K)
            IX1=INY(I1)*(2**(I11-1))+IX1
            IF(I11.NE.K) GO TO 130
            IF(IX1.EQ.IP) IZCNT=IZCNT+1
            IX1=0
130         CONTINUE
            DO 240 L=1,LIMIT
            IZC1=IZCNT
            IX1=ISOL(L,1)+ISOL(L,2)*2+ISOL(L,3)*4
            IX2=ISOL(L,4)+ISOL(L,5)*2+ISOL(L,6)*4
            IF ((IX1.GT.2*IP).OR.(IX2.GT.2*IP)) GO TO 240
            IF(IX1.EQ.IP) IZC1=IZC1+1
```

```
      IF(IX2.EQ.IP) IZC1=IZC1+1
      IF(IZC1.GT.IZLIM) GO TO 240
      DO 140 I1=1,K
      I=I1-1
C
C     COPY THE NEW SOL.
C
      IOUTY(IO1+I1)=ISOL(L,I1)
      IOUTY(IO2-4+I1)=ISOL(L,I1+K)
  140 CONTINUE
      IF (IT.GT.ITLIM) GO TO 200
      DO 160 J=1,N
      IX=0
      DO 150 IIL=1,K
      L1=IIL-1
      IX=IOUTY((J-1)*3+IIL)*(2**L1)+IX
  150 CONTINUE
      ID(J)=IX-IP
  160 CONTINUE
      IF(N.EQ.(2*ITLIM)) GO TO 180
C
C     VARY THE UNSOLVED ELEMENTS FROM -IP TO +IP
C
  170 CALL AUTOCO(N,LRATIO)
      ID(ITLIM)=ID(ITLIM)+1
      IF(ID(ITLIM).LE.IP) GO TO 170
      GO TO 190
  180 CALL AUTOCO(N,LRATIO)
  190 CONTINUE
C
      GO TO 240
C
C     PACK THE SOL. INTO IYSOL
C
  200 ICEND=ICEND+1
      IF (ICEND.GT.8000) STOP
      IZ=0
      DO 210 LL=1,33
  210 IZ=2*IZ+IOUTY(LL)
      IYSOL(ICEND,1)=IZ
      IZ=0
      DO 220 LL=34,66
  220 IZ=2*IZ+IOUTY(LL)
      IYSOL(ICEND,2)=IZ
      IZ=0
      DO 230 LL=67,99
  230 IZ=2*IZ+IOUTY(LL)
      IYSOL(ICEND,3)=IZ
  240 CONTINUE
  250 CONTINUE
      ICPTR=ICPTR+1
      IF (ICPTR.GT.IPEND) GO TO 260
      CALL UNPACK(IYSOL(ICPTR,1),IYSOL(ICPTR,2),
     1IYSOL(ICPTR,3))
      GO TO 60
  260 IF(IT.LE.ITLIM) GO TO 270
      IF(ICEND.EQ.IPEND) GO TO 270
      CALL UNPACK(IYSOL(ICPTR,1),IYSOL(ICPTR,2),
     1IYSOL(ICPTR,3))
      GO TO 50
  270 CONTINUE
```

```
  280   CONTINUE
        STOP
        END
        SUBROUTINE SOLVE
        COMMON ISOL(64,6),LIMIT,IRHS,ICOEFF(6),INOVAR,ID(
       1INY(99)
        DIMENSION IBAR(6),IY(6),INTER(6),ISAVE(6)

C       INOVAR IS NUMBER OF COEFFICIENTS BEING PASSED

C       SET FLAGS ANDSAVE COEFFICIENTS

        DO 10 I=1,INOVAR
        IBAR(I)=0
        ISAVE(I)=ICOEFF(I)
        IF (ICOEFF(I).GE.0) GO TO 10
        ICOEFF(I)=-ICOEFF(I)
        IRHS=IRHS+ICOEFF(I)
        IBAR(I)=1
   10   CONTINUE

C       FIND ALL SOLUTIONS

        LIMIT=0
        DO 100 I1=1,2
        IY(1)=0
        ITEMP1=0
        INTER(1)=0
        IF (I1.EQ.1) GO TO 30
        IY(1)=1
        ITEMP1=ITEMP1+ICOEFF(1)
        IF (ITEMP1.GT.IRHS) GO TO 100
        INTER(1)=ITEMP1
   30   CONTINUE
        DO 100 I2=1,2
        IY(2)=0
        ITEMP1=INTER(1)
        INTER(2)=INTER(1)
        IF (I2.EQ.1) GO TO 40
        IY(2)=1
        ITEMP1=ITEMP1+ICOEFF(2)
        IF (ITEMP1.GT.IRHS) GO TO 100
        INTER(2)=ITEMP1
   40   CONTINUE
        DO 100 I3=1,2
        IY(3)=0
        ITEMP1=INTER(2)
        INTER(3)=INTER(2)
        IF (I3.EQ.1) GO TO 50
        IY(3)=1
        ITEMP1=ITEMP1+ICOEFF(3)
        IF (ITEMP1.GT.IRHS) GO TO 100
        INTER(3)=ITEMP1
   50   CONTINUE
        DO 100 I4=1,2
        IY(4)=0
        ITEMP1=INTER(3)
        INTER(4)=INTER(3)
        IF (I4.EQ.1) GO TO 60
        IY(4)=1
        ITEMP1=ITEMP1+ICOEFF(4)
```

```
      IF (ITEMP1.GT.IRHS) GO TO 100
      INTER(4)=ITEMP1
   60 CONTINUE
      DO 100 I5=1,2
      IY(5)=0
      ITEMP1=INTER(4)
      INTER(5)=INTER(4)
      IF (I5.EQ.1) GO TO 70
      IY(5)=1
      ITEMP1=ITEMP1+ICOEFF(5)
      IF (ITEMP1.GT.IRHS) GO TO 100
      INTER(5)=ITEMP1
   70 CONTINUE
      DO 100 I6=1,2
      IY(6)=0
      ITEMP1=INTER(5)
      INTER(6)=INTER(5)
      IF (I6.EQ.1) GO TO 80
      IY(6)=1
      ITEMP1=ITEMP1+ICOEFF(6)
      IF (ITEMP1.GT.IRHS) GO TO 100
      INTER(6)=ITEMP1
   80 CONTINUE
      ITEMP=IRHS
      DO 90 I=1,INOVAR
      ITEMP=ITEMP-IY(I)*ICOEFF(I)
      IF (ITEMP.LT.0) GO TO 100
   90 CONTINUE
      IF (ITEMP.NE.0) GO TO 100
      LIMIT=LIMIT+1
      DO 100 I=1,INOVAR
      ISOL(LIMIT,I)=IY(I)
  100 CONTINUE

C     EXAMINE FLAGS AND COMPLEMENT IF REQUIRED AND RES
C     COEFFICIENTS

      DO 110 I=1,INOVAR
      ICOEFF(I)=ISAVE(I)
      IF (IBAR(I).EQ.0) GO TO 110
      DO 110 J=1,LIMIT
      ITEMP=ISOL(J,I)
      ISOL(J,I)=0
      IF (ITEMP.EQ.0) ISOL(J,I)=1
  110 CONTINUE
  120 RETURN
      END
      SUBROUTINE AUTOCO(NO,LRATIO)
      COMMON ISOL(64,6),LIMIT,IRHS,ICOEFF(6),INOVAR,ID(
     1INY(99)
      DIMENSION IB(32),IC(32)

C     AUTO CORRELATION
      DO 10 J=1,NO
      IB(J)=ID(J)
      IC(J)=0
   10 CONTINUE
      DO 30 I=1,NO
      K=NO+1-I
      L=0
      DO 20 J=1,I
```

```
      L1=K+L
      IC(J)=IC(J)+ID(K)*IB(L1)
      L=L+1
   20 CONTINUE
   30 CONTINUE
      MAXRAT=100000
      IBIG=IABS(IB(1))
      DO 35 I=2,ND
      IF (IC(I).EQ.0) GO TO 33
      IRATIO=IABS(IC(1)/IC(I))
      IF (IRATIO.LT.LRATIO) GO TO 50
      IF (IRATIO.LT.MAXRAT) MAXRAT=IRATIO
   33 IF (IABS(IB(I)).GT.IBIG) IBIG=IABS(IB(I))
   35 CONTINUE
      C=IC(1)
      DEN=ND*IBIG*IBIG
      EFF=C/DEN
      RATIO=MAXRAT
      PRINT40,(IB(L),L=1,ND),(IC(L1),L1=1,ND),EFF,RATIO
   40 FORMAT(1H0,5X,1H[,14I2,1H],5X,1H[,14I3,1H],F10.2,F10.2
   50 RETURN
      END
      SUBROUTINE UNPACK(IZ1,IZ2,IZ3)
      COMMON ISOL(64,6),LIMIT,IRHS,ICOEFF(6),INOVAR,ID(64),
     1INY(99)
      DO 10 LL=1,33
      LL2=34-LL
      IZ=IZ1
      IZ1=IZ1/2
   10 INY(LL2)=IZ-2*IZ1
      DO 20 LL=34,66
      LL2=100-LL
      IZ=IZ2
      IZ2=IZ2/2
   20 INY(LL2)=IZ-2*IZ2
      DO 30 LL=67,99
      LL2=166-LL
      IZ=IZ3
      IZ3=IZ3/2
   30 INY(LL2)=IZ-2*IZ3
      RETURN
      END
```

```
      C          THIS GENERATES INTEGER SEQUENCES MAG(C1,CN)=1
                  COMMON ISOL(256,8),LIMIT,TRHS,ICOEFF(8),INOVAR,
                  1ID(64),INY(132)
                  DIMENSION INSOL(8,16),IOUTY(132),IYSOL(8000,4),
                  1IA(16,400)
      C           ISOL(64,6) CONTAINS SOLUTIONS RETURNED BY SUB.
      C           SOLVE
      C           'LIMIT' IS THE NUMBER OF ABOVE SOLUTIONS
      C           'TRHS' IS THE RIGHT HAND SIDE OF BOOLEAN EQN.
      C           ICOEFF(6) CONTAINS COEFFICIENTS BEING
      C           PASSED TO SOLVE
      C           INOVAR IS THE NUMBER OF VARIABLES BEING PROCESSED
      C           IN SOLVE
      C           ID(64) CONTAINS A READY SOLUTION FOR FINDING
      C           AUTOCORELATION
      C           INY(99) CONTAINS THE INPUT VALUES FOR CURRENT
      C           SOLUTION
      C           INSOL(6,2) CONTAINS THE INITIALLY ASSUMED VALUES
      C           OF END COEFFICIENTS
      C           IOUTY(99) CONTAINS THE OUTPUT VALUES OF CURRENT
      C           SOLUTIONS
      C           IYSOL(8000,3) CONTAINS PACKED SOLUTIONS OF ONE
      C           ITERATIONS
      C           'IA' IS THE SPECIFIED GIVEN AUTOCORRELATION
                  READ*,N,K,IP,ENDITR,ITR,LRATIO,IZLIM
      C           N IS THE LENGTH OF SEQUENCE
      C           'K' IS THE NUMBER OF BINARY BITS REQD TO REPRESENT
      C           2*IP
      C           'IP'IS THE VALUE OF MAX PERMISSIBLE ELEMENT VALUE
      C           'ITR' IS THE NUMBER OF ITERATIONS
      C           'LRATIO' IS RATIO OF MAX.LOBE TO SIDE LOBES
      C           'IZLIM' IS THE MAX. NO. OF ZERO ELEMENTS ALLOWED
      C           IN A SEQUENCE
                  READ*,((INSOL(I,J),I=1,8),J=1,12)
                  READ*,((IA(I,J), I=1,N-1),J=1,ITR)
                  DO 280 INS=1,ENDITR
      C           TRANSFER INITIAL VALUES OF END COEFFCIENTS INTO
      C           'INY'
```

```
      DO 10 J=1,2*K
      K2=J
      IF(J.GT.K) K2=K2+K*N-8
      INY(K2)=INSOL(J,INS)
   10 CONTINUE

C     CALCULATE COEFFICIENTS TO BE PASSED TO SUBROUTINE
C     SOLVE

      DO 40 I1=1,K
      I=I1-1
      ICOEFF(K+I1)=0
      DO 20 I2=1,K
      I21=I2-1
      ICOEFF(K+I1)=ICOEFF(K+I1)+(2**I21)*INY(I2)
   20 CONTINUE
      ICOEFF(K+I1)=(ICOEFF(K+I1)-IP)*(2**I)
      ICOEFF(I1)=0
      DO 30 I2=1,K
      I21=I2-1
      IQ2=K*(N-1)+I2
      ICOEFF(I1)=ICOEFF(I1)+(2**I21)*INY(IQ2)
   30 CONTINUE
      ICOEFF(I1)=(ICOEFF(I1)-IP)*(2**I)
   40 CONTINUE
      DO 270 ITT=1,ITR
      ICOUNT=0
      IT=N-1
      INOVAR=8
      ICEND=1
C     ICEND IS A POINTER INDICATING AT ANY TIME THE END

C     OF ENTERIES IN IYSOL

      ICPTR=1

C     'ICPTR' IS THE POINTER TO CURRENT SOL.BEING
C     PROCESSED

      ITLIM=(N+1)/2
   50 IT=IT-1
      IPEND=ICEND

C     'IPEND' IS THE POINTER TO THE END OF ENTRIES IN
C     IYSOL DUE TO CURRENT SOL

C     CALCULATE 'IRHS'

   60 IRHS=IA(IT,ITT)-(N-IT)*IP*IP
      IRHS2=0
      DO 100 I1=1,K
      I=I1-1
      IRHS1=0
      IQ1=I+1
      IRHS1=IRHS1+IP*INY(IQ1)
      IF((N-IT).EQ.2)GO TO 90
      ISUM1=0
      DO 80 J=2,N-IT-1
      ISUM=0
      DO 70 IL=1,K
      IL1=IL-1
```

```
      IO1=(J-1)*K+1+IL1
      IO2=IO1+K*IT
      ISUM=ISUM+(2**IL1)*INY(IO2)
   70 CONTINUE
      IO1=(J-1)*K+1+I
      ISUM=ISUM*INY(IO1)
      IO2=IO1+K*IT
      ISUM=ISUM-IP*(INY(IO1)+INY(IO2))
      ISUM1=ISUM1+ISUM

   80 CONTINUE
      IRHS1=IRHS1-ISUM1
   90 J=N-IT

      IO1=(J-1)*K+1+I
      IO2=IO1+K*IT
      IRHS1=IRHS1+IP*INY(IO2)
      IRHS1=(2**I)*IRHS1
      IRHS2=IRHS2+IRHS1
  100 CONTINUE
      IRHS=IRHS+IRHS2
      CALL SOLVE
      ICOUNT=ICOUNT+1
  115 IF (LIMIT.EQ.0) GO TO 250
      IZCNT=0
      I11=0
      IX1=0
      IO1=(N-IT-1)*K

C     COPY PREVIOUS LEADING ELEMENTS

      DO 120 I1=1,IO1
      IOUTY(I1)=INY(I1)
      I11=I11+1
      IF(I11.GT.K) I11=MOD(I11,K)
      IX1=INY(I1)*(2**(I11-1))+IX1
      IF(I11.NE.K) GO TO 120
      IF(IX1.EQ.IP) IZCNT=IZCNT+1
      IX1=0
  120 CONTINUE
      I11=0
      IX1=0
      IO2=K*IT+K+1

C     COPY PREVIOUS FOLLOWING ELEMENTS

      DO 130 I1=IO2,K*N
      IOUTY(I1)=INY(I1)
      I11=I11+1
      IF(I11.GT.K) I11=MOD(I11,K)
      IX1=INY(I1)*(2**(I11-1))+IX1
      IF(I11.NE.K) GO TO 130
      IF(IX1.EQ.IP) IZCNT=IZCNT+1
      IX1=0
  130 CONTINUE
      DO 240 L=1,LIMIT
      IZC1=IZCNT
      IX1=ISOL(L,1)+ISOL(L,2)*2+ISOL(L,3)*4+ISOL(L,4)*8
      IX2=ISOL(L,5)+ISOL(L,6)*2+ISOL(L,7)*4+ISOL(L,8)*8
      IF ((IX1.GT.2*IP).OR.(IX2.GT.2*IP)) GO TO 240
      IF(IX1.EQ.IP) IZC1=IZC1+1
```

```
      IF(IX2.EQ.IP) IZC1=IZC1+1
      IF(IZC1.GT.IZLIM) GO TO 240
      DO 140 I1=1,K
      I=I1-1

C     COPY THE NEW SOL.

      IOUTY(IQ1+I1)=ISOL(L,I1)
      IOUTY(IQ2-K-1+I1)=ISOL(L,I1+K)
140   CONTINUE
      IF (IT.GT.ITLIM) GO TO 200
      DO 160 J=1,N
      IX=0
      DO 150 IIL=1,K
      L1=IIL-1
      IX=IOUTY((J-1)*K+IIL)*(2**L1)+IX
150   CONTINUE
      ID(J)=IX-IP
160   CONTINUE
      IF(N.EQ.(2*ITLIM)) GO TO 180

C     VARY THE UNSOLVED ELEMENTS FROM -IP TO +IP

170   CALL AUTOCO(N,LRATIO)
      ID(ITLIM)=ID(ITLIM)+1
      IF(ID(ITLIM).LE.IP) GO TO 170
      GO TO 190
180   CALL AUTOCO(N,LRATIO)
190   CONTINUE

      GO TO 240

C     PACK THE SOL. INTO IYSOL

200   ICEND=ICEND+1
      IF (ICEND.GT.8000) STOP
      IZ=0
      DO 210 LL=1,33
210   IZ=2*IZ+IOUTY(LL)
      IYSOL(ICEND,1)=IZ
      IZ=0
      DO 220 LL=34,66
220   IZ=2*IZ+IOUTY(LL)
      IYSOL(ICEND,2)=IZ
      IZ=0
      DO 230 LL=67,99
230   IZ=2*IZ+IOUTY(LL)
      IYSOL(ICEND,3)=IZ
      IZ=0
      DO 235 LL=100,132
235   IZ=2*IZ+IOUTY(LL)
      IYSOL(ICEND,4)=IZ
240   CONTINUE
250   CONTINUE
      ICPTR=ICPTR+1
      IF (ICPTR.GT.ICEND) GO TO 260
      CALL UNPACK(IYSOL(ICPTR,1),IYSOL(ICPTR,2),
     1IYSOL(ICPTR,3),IYSOL(ICPTR,4))
      GO TO 60
260   IF(IT.LE.ITLIM) GO TO 270
      CALL UNPACK(IYSOL(ICPTR,1),IYSOL(ICPTR,2),
```

```
      1,IYSOL(ICPTR,3),IYSOL(ICPTR,4))
      GO TO 50
270   CONTINUE
280   CONTINUE
      STOP
      END
      SUBROUTINE SOLVE
      COMMON ISOL(256,8),LIMIT,IRHS,ICOEFF(8),INOVAR,ID(64)
      1INY(132)
      DIMENSION IBAR(8),IY(8),INTER(8),ISAVE(8)

C     INOVAR IS NUMBER OF COEFFICIENTS BEING PASSED

C     SET FLAGS ANDSAVE COEFFICIENTS

      DO 10 I=1,INOVAR
      IBAR(I)=0
      ISAVE(I)=ICOEFF(I)
      IF (ICOEFF(I).GE.0) GO TO 10
      ICOEFF(I)=-ICOEFF(I)
      IRHS=IRHS+ICOEFF(I)
      IBAR(I)=1
10    CONTINUE

C     FIND ALL SOLUTIONS

      LIMIT=0
      DO 100 I1=1,2
      IY(1)=0
      ITEMP1=0
      INTER(1)=0
      IF (I1.EQ.1) GO TO 30
      IY(1)=1
      ITEMP1=ITEMP1+ICOEFF(1)
      IF (ITEMP1.GT.IRHS) GO TO 100
      INTER(1)=ITEMP1
30    CONTINUE
      DO 100 I2=1,2
      IY(2)=0
      ITEMP1=INTER(1)
      INTER(2)=INTER(1)
      IF (I2.EQ.1) GO TO 40
      IY(2)=1
      ITEMP1=ITEMP1+ICOEFF(2)
      IF (ITEMP1.GT.IRHS) GO TO 100
      INTER(2)=ITEMP1
40    CONTINUE
      DO 100 I3=1,2
      IY(3)=0
      ITEMP1=INTER(2)
      INTER(3)=INTER(2)
      IF (I3.EQ.1) GO TO 50
      IY(3)=1
      ITEMP1=ITEMP1+ICOEFF(3)
      IF (ITEMP1.GT.IRHS) GO TO 100
      INTER(3)=ITEMP1
50    CONTINUE
      DO 100 I4=1,2
      IY(4)=0
      ITEMP1=INTER(3)
      INTER(4)=INTER(3)
```

```
      IF (I4.EQ.1) GO TO 60
      IY(4)=1
      ITEMP1=ITEMP1+ICOEFF(4)
      IF (ITEMP1.GT.IRHS) GO TO 100
      INTER(4)=ITEMP1
   60 CONTINUE
      DO 100 I5=1,2
      IY(5)=0
      ITEMP1=INTER(4)
      INTER(5)=INTER(4)
      IF (I5.EQ.1) GO TO 70
      IY(5)=1
      ITEMP1=ITEMP1+ICOEFF(5)
      IF (ITEMP1.GT.IRHS) GO TO 100
      INTER(5)=ITEMP1
   70 CONTINUE
      DO 100 I6=1,2
      IY(6)=0
      ITEMP1=INTER(5)
      INTER(6)=INTER(5)
      IF (I6.EQ.1) GO TO 80
      IY(6)=1
      ITEMP1=ITEMP1+ICOEFF(6)
      IF (ITEMP1.GT.IRHS) GO TO 100
      INTER(6)=ITEMP1
   80 CONTINUE
      DO 100 I7=1,2
      IY(7)=0
      ITEMP1=INTER(6)
      INTER(7)=INTER(6)
      IF (I7.EQ.1) GO TO 81
      IY(7)=1
      ITEMP1=ITEMP1+ICOEFF(6)
      IF (ITEMP1.GT.IRHS) GO TO 100
      INTER(7)=ITEMP1
   81 CONTINUE
      DO 100 I8=1,2
      IY(8)=0
      ITEMP1=INTER(7)
      INTER(8)=INTER(7)
      IF (I8.EQ.1) GO TO 82
      IY(8)=1
      ITEMP1=ITEMP1+ICOEFF(7)
      IF (ITEMP1.GT.IRHS) GO TO 100
      INTER(8)=ITEMP1
   82 CONTINUE
      ITEMP=IRHS
      DO 90 I=1,INOVAR
      ITEMP=ITEMP-IY(I)*ICOEFF(I)
      IF (ITEMP.LT.0) GO TO 100
   90 CONTINUE
      IF (ITEMP.NE.0) GO TO 100
      LIMIT=LIMIT+1
      DO 100 I=1,INOVAR
      ISOL(LIMIT,I)=IY(I)
  100 CONTINUE
C
C     EXAMINE FLAGS AND COMPLEMENT IF REQUIRED AND RESTORE
C     COEFFICIENTS
C
      DO 110 I=1,INOVAR
```

```
      ICOEFF(I)=ISAVE(I)
      IF (IBAR(I).EQ.0) GO TO 110
      DO 110 J=1,LIMIT
      ITEMP=ISOL(J,I)
      ISOL(J,I)=0
      IF (ITEMP.EQ.0) ISOL(J,I)=1
  110 CONTINUE
  120 RETURN
      END
      SUBROUTINE AUTOCO(NO,LRATIO)
      COMMON ISOL(256,8),LIMIT,IRHS,ICOEFF(8),INOVAR,ID(64),
     1INY(132)
      DIMENSION IB(32),IC(32),Y(32),Z(32)

C     AUTO CORRELATION

      DO 10 J=1,NO
      IB(J)=ID(J)
      IC(J)=0
   10 CONTINUE
      DO 30 I=1,NO
      K=NO+1-I
      L=0
      DO 20 J=1,I
      L1=K+L
      IC(J)=IC(J)+ID(K)*IB(L1)
      L=L+1
   20 CONTINUE
   30 CONTINUE
      MAXRAT=100000
      IBIG=IABS(IB(1))
      DO 35 I=2,NO
      IF (IC(I).EQ.0) GO TO 33
      IRATIO=IABS(IC(1)/IC(I))
      IF (IRATIO.GT.LRATIO) GO TO 50
      IF (IRATIO.GT.MAXRAT) MAXRAT=IRATIO
   33 IF (IABS(IB(I)).GT.IBIG) IBIG=IABS(IB(I))
   35 CONTINUE
      C=IC(1)
      DEN=NO*IBIG*IBIG
      EFF=C/DEN
      RATIO=MAXRAT
   36 CONTINUE
      PRINT40,(IB(L),L=1,NO),(IC(L1),L1=1,NO),EFF,RATIO
   40 FORMAT(1H0,5X,1H[,4I3,1H],5X,1H[,4I3, 1H],F8.0,F8.2)
   50 RETURN
      END
      SUBROUTINE UNPACK(IZ1,IZ2,IZ3,IZ4)
      COMMON ISOL(256,8),LIMIT,IRHS,ICOEFF(8),INOVAR,ID(64)
     1INY(32)
      DO 10 LL=1,33
      LL2=34-LL
      IZ=IZ1
      IZ1=IZ1/2
   10 INY(LL2)=IZ-2*IZ1
      DO 20 LL=34,66
      LL2=100-LL
      IZ=IZ2
      IZ2=IZ2/2
   20 INY(LL2)=IZ-2*IZ2
      DO 30 LL=67,99
```

```
      LL2=165-LL
      IZ=IZ3
      IZ3=IZ3/2
   10 IDY(LL2)=IZ-2*IZ3
      DO 40 LL=100,132
      LL2=232-LL
      IZ=IZ4
      IZ4=IZ4/2
   40 IDY(LL2)=IZ-2*IZ4
      RETURN
      END
```

```
C       THIS PROGRAM ESTIMATES IMPULSE RESPONSE
C       IT USES PN SEQUENCES

        DIMENSION C(8,32),H(32),HDASH(512,32),Y(64),
       1HDASH1(32),SNR(4),PCERROR(32),ERROR(32),WN(64)
        READ*,N1,N2,IAVR,N
        READ*,(SNR(IN),IN=1,4)
        READ*,((C(J,I),I=1,N1), J=1,N)
        READ*,(H(I),I=1,N2)
        DATA PI/3.14159265/
        PI2=2.*PI
        PRINT 220,(H(I),I=1,N2)
220     FORMAT(1H0,'GIVEN RESP =',10X,10F8.2)
        DO 70 JJJ=1,N
        DO 65 IN=1,4
        PRINT 100,N1,SNR(IN),(C(JJJ,I),I=1,N1)
100     FORMAT (1H ,10X,I6,F6.1,10X,31F5.1)
        PRINT 140
140     FORMAT(1H0,10X,'EST RESP',50X,'ERROR')
        DO 60 KK=1,IAVR
        IIQ=KK-1
        IQ=2**IIQ
        C1=0
        DO 5 I=1,N1
        C1=C1+C(JJJ,I)*C(JJJ,I)
5       CONTINUE

C       CONVOLUTING INPUT SIGNAL WITH SYSTEM RESP.

        DO 20 K=1,N1+N2-1
        SUM1=0
        DO 15 J=1,N2
        IF((K-J+1).GT.N1) GO TO 15
        IF ((K-J+1).LT.1) GO TO 15
        SUM1=SUM1+C(JJJ,K-J+1)*H(J)
15      CONTINUE
        Y(K)=SUM1
20      CONTINUE

C       CALCULATING AV POWER IN OUTPUT SIGNAL

        SUM=0
        DO 22 I=1,N1+N2-1
        SUM=SUM+Y(I)**2
22      CONTINUE
        AVRPY=SQRT(SUM/(N1+N2-1))

C       GENRATING NOISE

        SIGMA=SQRT(10.**(-0.1*SNR(IN)))
        DO 40 MM=1,IQ
        DO 25 I=1,N1+N2-1
        R2=RAN(DUM)
        R3=RAN(DUM)
        WN(I)=SIGMA*(SQRT(-2.*ALOG(R2)))*COS(PI2*R3)
       1/AVRPY
25      CONTINUE

C       ESTIMATING SYSTEM RESPONSE

        DO 40 K=1,N2
```

```
      SUM=0
      DO 30 J=K,N1+N2-1
      J1=J-K
      J2=MOD((J1),N1)+1
      SUM=SUM+C(JJJ,J2)*(Y(J)+WN(J))
30    CONTINUE
      IF(K.EQ.1) GO TO 36
      DO 35 J=1,K-1
      J1=N1-K+J+1
      SUM=SUM+C(JJJ,J1)*(Y(J)+WN(J))
35    CONTINUE
36    HDASH(MM,K)=SUM/C1
40    CONTINUE
      DO 48 I=1,N2
      AVR=IQ
      SUM=0
      DO 45 LL=1,IQ
      SUM=SUM+HDASH(LL,I)
45    CONTINUE
      HDASH1(I)=SUM/AVR
48    CONTINUE

C     CALCULATING ERROR AND PERCENT ERROR

      DO 50 J=1,N2
      ERROR(J)=(H(J)-HDASH1(J))
      PCEROR(J)=ERROR(J)*100/H(J)
50    CONTINUE

C     CALCULATING MEAN SQ ERROR,MAX ERROR

      SUM=0
      DO 52 I=1,N2
      SUM=SUM+(ERROR(I))**2
52    CONTINUE
      ZMSER=SQRT(SUM/10.)
      BIG=ABS(ERROR(1))
      DO 54 I=2,N2
      IF(BIG.LT.ABS((ERROR(I))))BIG=ABS(ERROR(I))
54    CONTINUE
      ZMAXER=BIG
      PRINT 110,IQ
110   FORMAT (1H ,3X,I6)
      PRINT 200,(HDASH1(I),I=1,N2),(ERROR(I),I=1,N2)
200   FORMAT(1H0,2X,1H[,10F7.2,1H],2X,1H[,10F7.2,1H])
      PRINT 205, (PCEROR(I),I=1,N2)
205   FORMAT(1H0,30X,'PERCENT ERROR =     ',10X,
     11H[,10F7.2,1H])
      PRINT 210,ZMAXER,ZMSER
210   FORMAT(1H0,'MAX ERROR='F8.2,20X,'MEAN SQ
     1 ERROR='F8.2)
60    CONTINUE
65    CONTINUE
70    CONTINUE
      STOP
      END
```

APPENDIX N

```
C     THIS PROGRAM ESTIMATES IMPULSE RESPONSE
C     IT USES BARKER SEQUENCES
C
      DIMENSION C(8,32),H(32),HDASH(16,32),Y1(64),
     1HDASH1(128,32),Y2(64),H1(32),HDASH2(32),
     1ERROR(32),HN(32),SNR(4),PCEROR(32)
      READ*,N1,N2,IAVR,N
      READ*,(SNR(I),I=1,4)
      READ*,((C(J,I),I=1,N1), J=1,N)
      READ*,(H(I),I=1,N2)
      PRINT 220,(H(I),I=1,N2)
  220 FORMAT(1H0,'GIVEN RESP =',10X,10F8.2)
      DATA PI/3.14159265/
      PI2=2.*PI
      DO 70 JJJ=1,N
      DO 65 IN=1,4
      SIGMA=SQRT(10.**(-0.1*SNR(IN)))
      PRINT 100,N1,SNR(IN),(C(JJJ,I),I=1,N1)
  100 FORMAT (1H ,10X,I6,F6.1,10X,13F6.1)
      PRINT 140
  140 FORMAT(1H0,10X,'EST RESP',50X,'ERROR')
      DO 60 KK=1,IAVR
      IID=KK-1
      ID=2**IID
      C1=0
      DO 5 I=1,N1
      C1=C1+C(JJJ,I)*C(JJJ,I)
    5 CONTINUE
      DO 46 LL=1,ID
      DO 40 MM=1,N2+1
      DO 10 I=1,N2
      H1(I)=H(I)
   10 CONTINUE
      IF(MM.EQ.1)GO TO 11
      H1(MM-1)=0
C
C     CONVOLUTING INPUT SIGNAL WITH SYSTEM RESP.
C
   11 DO 20 K=1,N1+N2-1
      SUM1=0
      SUM2=0
      DO 15 J=1,N2
      IF((K-J+1).GT.N1) GO TO 15
      IF ((K-J+1).LT.1) GO TO 15
      SUM1=SUM1+C(JJJ,K-J+1)*H1(J)
   15 CONTINUE
      Y1(K)=SUM1
   20 CONTINUE
C
C     CALCULATING AV POWER IN OUTPUT SIGNAL
C
      SUM=0
      DO 22 I=1,N1+N2-1
      SUM=SUM+Y1(I)**2
   22 CONTINUE
C
C     GENRATING NOISE
C
      AVRPY=SQRT(SUM/(N1+N2-1))
      DO 25 I=1,N1+N2-1
```

```
      R2=RAN(DUM)
      R3=RAN(DUM)
      WN(I)=SIGMA*(SQRT(-2.*ALOG(R2)))*COS(PI2*R3)
     1/AVDPY
   35 CONTINUE
C
C     ESTIMATING SYSTEM RESPONSE
C
      DO 40 K=1,N2
      SUM=0
      DO 30 J=1,N1
      SUM=SUM+C(IJJ,J)*(Y1(K+J-1)+WN(K+J-1))/C1
   30 CONTINUE
      HDASH(MM,K)=SUM
   40 CONTINUE
      IM=1
      DO 45 J=1,N2+1
      SUM=0
      SUM=SUM+HDASH(1,J)-HDASH(IM+1,J)
      IM=IM+1
      HDASH1(LL,I)=SUM
   45 CONTINUE
   46 CONTINUE
      DO 48 I=1,N2
      AVR=IO
      SUM=0
      DO 47 LL=1,IO
      SUM=SUM+HDASH1(LL,I)
   47 CONTINUE
      HDASH2(I)=SUM/AVR
   48 CONTINUE
C
C     CALCULATING ERROR AND PERCENT ERROR
C
      DO 50 J=1,N2
      ERROR(J)=(H(J)-HDASH2(J))
      PCERROR(J)=ERROR(J)*100/H(J)
   50 CONTINUE
C
C     CALCULATING MEAN SQ ERROR,MAX ERROR
C
      SUM=0
      DO 52 I=1,N2
      SUM=SUM+(ERROR(I))**2
   52 CONTINUE
      ZMSER=SQRT(SUM/10.)
      BIG=ABS(ERROR(1))
      DO 54 I=2,N2
      IF(BIG.LT.ABS((ERROR(I))))BIG=ABS(ERROR(I))
   54 CONTINUE
      ZMAXER=BIG
      PRINT 110,IO
  110 FORMAT (1H ,3X,I6)
      PRINT 200,(HDASH2(I),I=1,N2),(ERROR(I),I=1,N2)
  200 FORMAT(1H0,2X,1H[,10F7.2,1H],2X,1H[,10F7.2,1H])
      PRINT 205, (PCERROR(I),I=1,N2)
  205 FORMAT(1H0,32X,'PERCENT ERROR =    ',10X,
     11H[,10F7.2,1H])
      PRINT 210,ZMAXER,ZMSER
  210 FORMAT(1H0,'MAX ERROR='F8.2,20X,'MEAN SQ
```

```
                  1ERROR='F8.2)
       60         CONTINUE
       65         CONTINUE
       70         CONTINUE
                  STOP
                  END
```

```
C         THIS PROGRAM ESTIMATES IMPULSE RESPONSE
C          IT USES INTEGER HUFMAN SEQUENCES

          DIMENSION C(8,32),H(32),HDASH(128,32),Y1(64),
          1HDASH1(32),Y2(64),SNR(4),PCERDR(64),ERROR(32),
          1WN(32)
          READ*,N1,N2,IAVR,N
          READ*,(SNR(I),I=1,4)
          READ*,((C(J,I),I=1,N1), J=1,N)
          READ*,(H(I),I=1,N2)
          PRINT 220,(H(I),I=1,N2)
220       FORMAT(1H0,'GIVEN RESP=',5X,10F8.2)
          DATA PI/3.14159265/
          PI2=2.*PI
          DO 70 JJJ=1,N
          DO 65 IN=1,4
          PRINT 100,N1,SNR(IN),(C(JJJ,I),I=1,N1)
100       FORMAT (1H ,10X,I6,F6.1,10X,11F6.2)
          PRINT 140
140       FORMAT (1H0,15X,'EST.RESP',50X,'ERROR')
          DO 60 KK=1,IAVR
          IIO=KK-1
          IO=2**IIO
          C1=0
          DO 5 I=1,N1
          C1=C1+C(JJJ,I)*C(JJJ,I)
5         CONTINUE

C         CONVOLUTING INPUT SIGNAL WITH SYSTEM RESP.

          DO 20 K=1,N1+N2-1
          SUM1=0
          SUM2=0
          DO 15 J=1,N2
          IF((K-J+1).GT.N1) GO TO 15
          IF ((K-J+1).LT.1) GO TO 15
          SUM1=SUM1+C(JJJ,K-J+1)*H(J)
15        CONTINUE
          Y1(K)=SUM1
20        CONTINUE

C         CALCULATING AV POWER IN OUTPUT SIGNAL

          SUM=0
          DO 22 I=1,N1+N2-1
          SUM=SUM+Y1(I)**2
22        CONTINUE

C         GENRATING NOISE

          AVRPY=SQRT(SUM/(N1+N2-1))
          SIGMA=SQRT(10.**(-0.1*SNR(IN)))
          DO 40 MM=1,IO
          DO 25 I=1,N1+N2-1
          R2=RAN(DUM)
          R3=RAN(DUM)
          WN(I)=SIGMA*(SQRT(-2.*ALOG(R2)))*COS(PI2*R3)
          1/AVRPY
25        CONTINUE

C         ESTIMATING SYSTEM RESPONSE
```

```
      DO 40 K=1,N2
      SUM=0
      DO 30 J=1,N1
20    SUM=SUM+C(III,J)*(Y1(K+J-1)+WN(K+J-1))/C1
30    CONTINUE
      HDASH(MM,K)=SUM
40    CONTINUE
      DO 48 I=1,N2
      AVR=IO
      SUM=0
      DO 45 LL=1,IO
      SUM=SUM+HDASH(LL,I)
45    CONTINUE
      HDASH1(I)=SUM/AVR
48    CONTINUE

C     CALCULATING ERROR AND PERCENT ERROR

      DO 50 J=1,N2
      ERROR(J)=(H(J)-HDASH1(J))
      PCEROR(J)=ERROR(J)*100/H(J)
50    CONTINUE

C     CALCULATING MEAN SQ ERROR,MAX ERROR

      SUM=0
      DO 52 I=1,N2
      SUM=SUM+(ERROR(I))**2
52    CONTINUE
      ZMSER=SQRT(SUM/10.)
      BIG=ABS(ERROR(1))
      DO 54 I=2,N2
      IF(BIG.LT.ABS((ERROR(I))))BIG=ABS(ERROR(I))
54    CONTINUE
      ZMAXER=BIG
      PRINT 110,IO
110   FORMAT (1H ,3X,I6)
      PRINT 200,(HDASH1(I),I=1,N2)/(ERROR(I),I=1,N2)
200   FORMAT(1H0,2X,1H[,10F7.2,1H],2X,1H[,10F7.2,1H])
      PRINT 205, (PCEROR(I),I=1,N2)
205   FORMAT(1H0,35X,'PERCENT ERROR =    ',10X,
     11H[,10F7.2,1H])
      PRINT 210,ZMAXER,ZMSER
210   FORMAT(1H0,'MAX ERROR='F8.2,20X,'MEAN SQ
     1 ERROR='F8.2)
60    CONTINUE
65    CONTINUE
70    CONTINUE
      STOP
      END
```